PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06K — A2

(11) International Publication Number: **WO 98/08180**

(43) International Publication Date: 26 February 1998 (26.02.98)

(21) International Application Number: PCT/IL97/00266

(22) International Filing Date: 5 August 1997 (05.08.97)

(30) Priority Data:

08/689,209	5 August 1996 (05.08.96)	US
60/038,080	6 March 1997 (06.03.97)	US

(71) Applicant: T.T.R. TECHNOLOGIES LTD. [IL/IL]; Hanagar Street 2, 44425 Kfar-Saba (IL).

(72) Inventors: SOLLISH, Bruce, David; Beit Israel Street 43, 44854 Emmanuel (IL). HOWE, Dennis; 6141 N. Paseo Valdear, Tucson, AZ 85750 (US). ISRAEL, Henry, Marshall; Ben Zackai Street 39, 51482 Bnei Brak (IL).

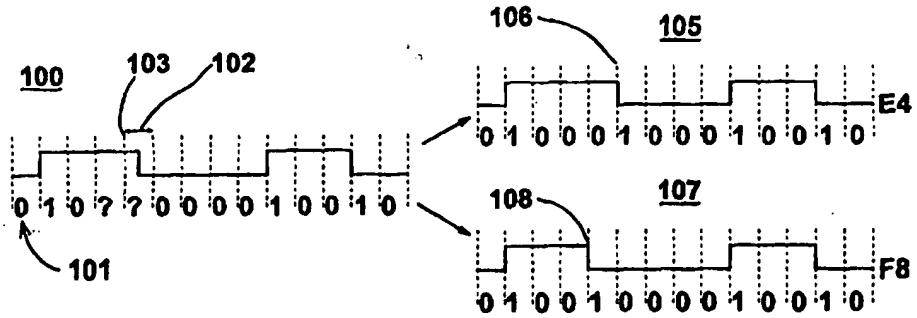
(74) Agent: A. TALLY EITAN - ZEEV PEARL, D. LATZER & CO.; Law Offices, Lumir House, Maskit Street 22, 46733 Herzelia (IL).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, UZ, VN, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published

Without international search report and to be republished upon receipt of that report.

(54) Title: DIGITAL OPTICAL MEDIA AUTHENTICATION AND COPY PROTECTION METHOD



(57) Abstract

Novel digital optical media has recorded thereon certain symbols belonging to two classes of non-standard codes in precise predetermined locations. One class provides symbols which, when read many times by a standard optical media reader, are decoded as valid but having variable values. A second class embodies codes which are immediately recognized by the player's decoder as invalid. The first class of non-standard codes can be read by a standard optical media reader but cannot be written or reproduced by standard optical media recorders and mastering equipment, and its presence on optical media thereby serves to identify the optical media as authentic, as opposed to an unauthorized copy, which will lack these special symbols. Symbols belonging to the second class of non-standard codes serve to protect the reading of symbols belonging to the first class from being altered or stabilized by the error-correcting system of the player. Patterns combining symbols of these two classes provide a non-copyable mark for automatically verifying the authenticity of optical media and protecting the data recorded thereon from being usable except when present on authentic media.

BN

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

DIGITAL OPTICAL MEDIA AUTHENTICATION AND COPY PROTECTION METHOD

FIELD AND BACKGROUND OF THE INVENTION

The present invention relates to data protection in general and to protection of data on optical media, such as Compact Disc Digital Audio (CDDA), Compact Disc Read-Only Memory (CD-ROM), and Digital Video Disc (DVD). The present application utilizes the term "digital optical media" to denote any media which stores data in digital form which can be read by optical means, including but not limited to laser. The present application utilizes the term "data" to refer to any information stored on media, including, but not limited to text, images, program instructions and parameters (collectively known as "software"), audio, and video, including interactive graphics and motion pictures.

Digital optical media is well-known in the art and is utilized to store large amounts of data in digital form, such as audio data, video data, software data, or document data. Software, document, and audio-visual data ("multi-media" data) may be read and utilized by a computer from digital optical media, such as Compact Disc Read-Only Memory (CD-ROM). There are also widely-available players for reading data from digital optical media and using this data to reconstruct audio, visual, text, and audio-visual information. Such players include, but are not limited to, CD players, CD-ROM multi-media players, game-playing systems, and DVD-players, which can reproduce sound, images, text, and motion pictures from data stored on digital optical media. Some computers are also configured to duplicate the functionality of CD players, CD-ROM multi-media players, game-playing systems, and DVD-players.

In order to protect the proprietary rights of the creators or owners of the data recorded on media, it is desirable to be able to determine if a particular instance of media is authentic or if it is an unauthorized copy. It is furthermore desirable to be able to effect such a determination in an automatic way, such as by computer without having to rely on a physical inspection of the media. It is also

desirable to be able to prevent the copying of data stored on media, or, equivalently, to render an unauthorized copy of the data unusable. The present application will utilize the term "authentication" to denote determining whether an instance of media is original as opposed to an unauthorized copy, and will utilize the term "copy protection" to refer to any method which prevents unauthorized copying of data, which renders unusable an unauthorized copy of that data, or which permits only authorized copies of the data to be usable.

In the case of optical media, small-quantity duplication of data is possible with an optical media recorder, such as a CD-ROM recorder. Low-cost optical media recorders are generally available to the public, and it is both easy and inexpensive for an ordinary consumer who has access to such a device to duplicate the data on a piece of optical media, in violation of the copyrights or other proprietary rights which may exist for the data on such media. Large-quantity duplication requires a mastering machine, a technology which is expensive and not generally available. Certain commercial enterprises, however, duplicate optical media on a mass scale in violation of the copyrights or other proprietary rights which may exist for data on such media. Although such duplication requires special equipment, it results in a large quantity of unauthorized copies. Both small- and large-scale unauthorized copying deprive the creators and owners of data stored on media of control of the distribution and use of their property. There is therefore a widely-recognized need for, and it would be advantageous to have, an automatic method of verifying the authenticity of a particular instance of optical media, and a method of preventing the copying of data stored on optical media both on a small scale by consumers and on a large scale by commercial enterprises.

Description of Prior Art Digital Optical Media

Digital optical media technology is established according to a series of international standards, all of which are incorporated herein by reference, and are collectively referred to hereinafter as "standards". For example, some specific common standards applicable to CD's include: the International Standards

Organization (ISO) standard 9660 entitled "Information Processing — Volume and File Structure of CD-ROM for Information Interchange, ISO standard 13490-1", the International Electrotechnique Commission (CEI-IEC) standard 908 (also known as the "Red Book"), and ISO/IEC 10140 (also known as the "Yellow Book").

Fig 1 is a cross-sectional schematic of a portion of the data surface of a digital optical medium. Referring briefly to Fig. 1, according to these standards, illustrates digital optical media having at least one layer of transparent refractive material **10** which has data recorded on one surface which is coated with a reflective material **12**, and covered with an optional protective layer **14**. Reflective material **12** in combination with transparent refractive material **10** produces transparent reflective layer **24** whose optical properties depend on the properties both of reflective material **12** and transparent refractive material **10**. Transparent reflective layer **24** comprises the data surface (*i.e.* the surface on which optically detectable features that correspond to digital data reside) of the digital optimal medium. Layer **24** usually resides on one surface of a thick transparent disc-shaped substrate (substrate thickness is 1.2 mm for CD and 0.6 mm for DVD). Layer **24** is embossed directly onto a surface of such disc-shaped substrate when mass-producing digital optical media. Different embodiments of digital optical media have different numbers of transparent reflective layers **24**. CD's for example, have a single such layer, while DVD's have up to four such layers (two on each side of the disc substrate). Within transparent reflective layer **24** there are regions **16** having different optical properties such that their reflectivity varies significantly from one region to the next. A region **18** of high reflectivity will retro-reflect most of the incident light passing into transparent refractive layer **10**, whereas a region **20** of low reflectivity will absorb or scatter most of the incident light passing into transparent reflective layer **10**. Different embodiments of digital optical media utilize different optical principles to achieve this difference in reflectivity. In digital optical media that is mass-produced by plastic molding apparatus, for example, the difference arises from changes made in the local thickness of transparent refractive layer **10**, whereas for digital optical

media that are recorded individually by a computer-controlled optical recorder (*e.g.*, write-once media such as CD-R disks), the difference arises from bulk changes to the refractive index of transparent refractive layer **10**.

Reading Data from Digital Optical Media

In order to read the data written onto digital optical media, such as CD and DVD, the media is rotated at a precisely-controlled speed, and light from a laser is focused through the disc-shaped substrate into transparent reflective layer **24** from which it is reflected back to a detector which measures the intensity of the reflected light. During the recording or manufacturing process of the digital optical media, the optical properties of the layer **24** are physically modified according to the data to be recorded so that the reflected light will vary significantly in intensity depending on where the laser light strikes. Typically, there are two different intensity levels for the reflected light. A region **18** which reflects a high intensity of the laser light is referred to as "land", and a region **20** which reflects a low intensity of light is referred to as "pit". Pits and lands may be physically implemented in different ways, but they always have the property of reflecting discernibly different light intensities. Moreover, pits and lands are specified by the standards to have sharp, well-defined boundaries **22**, so that according to the standards it is normally possible to precisely identify the location where a pit ends and a land begins and where a land ends and a pit begins. The present application utilizes the term "edge" to refer to a precise, well-defined boundary between one region and another, as for example, boundary **22**.

Data is recorded onto digital optical media in a spiral track along which these patterns of pits and lands are laid out in a linear fashion. As the media spins, the laser light sweeps along the track and its reflected intensity depends on whether the light falls on land (high reflectivity) or pit (low reflectivity). A change in the reflected intensity is referred to as a "transition", and whenever the intensity of the reflected light changes from one value to another, (that is, when the incident light passes either from land to pit or from pit to land), the detector circuitry signals that a transition has occurred. It is not the intensity of the reflected light, however,

but rather the precise timing of these transitions from one intensity to the other (relative to a data clock maintained within the digital data detector of the medium reader) which represents the digital data recorded on the media. The standards imply that a detected transition will indicate the position of an edge.

Data Representation

Digital data is represented within a computer or digital optical media player as a series of "bits" (binary digits, *i.e.*, 1's and 0's), where 8 bits are typically grouped into a data unit referred to as a "byte". In general, the sequence of bits in unconstrained in the sense that any specific bit can be succeeded by a 1 or a 0. It is not desirable, however, to record unconstrained data on digital optical media using the recording technique previously described (*e.g.*, if pits represent 1's and lands represent 0's or if a transition occurs only when a 1 is recorded), because transitions may then occur too frequently or not frequently enough, depending on the data. For example, a long sequence of 1's or a long sequence of 0's would result in a very long space between transitions, and this would cause the data decoder clock to lose synchronization with the data recorded on the track. Moreover, on extremely long runs of 1's or 0's would cause a very long space to occur between successive pits in the in-track direction, this could interfere with the ability of the playback spot to follow such a track. A series of alternating 1's and 0's on the other hand would result in a very short space between transitions and would require the media reader to have an extremely small focused spot size. To avoid these problems, therefore, digital optical media standards specify a bi-directional mapping between bytes of the data stream and representations thereof, known as "symbols", recorded on the media. The mapping is bi-directional since any given data byte can be mapped into a unique symbol, which in turn can be mapped uniquely back into the original data byte. That is, these two directional mappings of the bi-directional mapping are inverses of one another. The specific mappings of digital optical media according to the present standards are such that prior to recording, every byte of data is encoded to convert it to a constrained binary sequence that exhibits at least a desired

minimum number of 0's, but not more than a desired maximum number of 0's, between any two 1's. The inverse directional mapping converts these bit sequences back into the original data bytes when the digital optical media is read. The constraints which specify the minimum and maximum number of consecutive 0's are known as "run length-limited rules", or "RLL rules".

This is illustrated as follows for the special case of a CD, where data bytes are converted to a 14-bit constrained sequence using a bi-directional mapping known as the Eight-to-Fourteen-Modulation (EFM) code, as is partially illustrated by way of example in the table of Fig. 2, to which reference is now briefly made. (Other forms of optical media, such as DVD, use a similar, but not identical bi-directional mapping.) The table of Fig. 2 has two columns, referenced 26 and 28, which list the byte values and the corresponding channel bit EFM codes, respectively. Each 14-bit EFM code sequence observes strict limits in the spacing of the transitions along the digital optical media data track. In the EFM code sequences, transitions are indicated by 1's, and no variation of the media track feature (*i.e.*, pit or land) is indicated by 0's, but only certain patterns are used. The defined EFM code sequences embodied on the CD have the property that region lengths are all integer multiples of an elementary length unit. Furthermore, edges occur no closer than three (3) elementary length units from one another, and no further than eleven (11) elementary length units from another. The value of an elementary length unit which corresponds to a single EFM code bit, may vary from one embodiment to another, but in CD digital optical media it is nominally on the order of 0.3 μm (micrometers). There are 256 different valid EFM codes which have been arbitrarily assigned to represent the 256 different byte patterns, and it is the EFM code sequences which are actually recorded on the digital optical media data track. The individual bits of these code sequences are referred to as "channel bits" of the recorded data, of which EFM encoding is but one embodiment of channel coding. A complete channel bit sequence representing a byte of data is known as a "symbol". For a CD, this is a 14-channel bit EFM code sequence.

Referring briefly to Fig. 3 which is a schematic illustration of digital signals, the player detects transitions and indicates them by a pulse **30** in time, and it indicates an absence of transition by a constant signal value **32**.This pulse signal can be obtained by taking a rectified derivative of signal **36** which is output by the disc player as its focused read spot scans the data track segment formed by pits **36**. When the signal is plotted as an ordinate **38** against a time abscissa divided into suitable time units **36**, which correspond to the scanning of a single elementary length unit (channel bit) along the optical media track. The positions of the transitions **30** indicate the channel bit 1's, and the constant signal positions indicate the channel bit 0's. In the example of Fig. 3, the detected 14-bit EFM code sequence 10001000100000. Referring to the table in Fig. 2 shows that the byte encoded by this particular symbol has a byte value 03.

To further insure that the minimum and maximum length limits that separate transitions are strictly observed, successive 14-bit EFM code sequences are joined by special 3-bit groups known as "merge bits" which contain no information, but are able to produce a transition if needed to maintain the transition spacing constraints. The use of such encoding as EFM places reasonable bounds on the frequency spectrum of the playback signal **35** regardless of the data recorded along the track and enables digital data to be read from the media with sufficient accuracy. When the player reads the digital optical media track, the timing of the channel bit transitions is measured relative to the period of the data detectors internal clock) to determine which code sequence is present, and this is then translated by a look-up table into the corresponding data byte value.

Physically, the channel bits are represented in the digital optical media as regions of pits alternating with regions of lands, such that the regions have well-defined sizes in the in-track direction which are integer multiples of a size corresponding to the length unit, a channel bit. As stated previously, for CD, this size is about 0.3 μm, and the minimum size of a region is three times this size (for

a total length of about 0.9 μm), while the maximum size of a region is eleven times this size (for a total length of about 3.3 μm).

Frames

In CD recording, EFM encoded symbols are used to build groups of data units called "frames", through which higher-level data organization and coordination is achieved. By way of example, FIG 4, to which reference is now briefly made, illustrates a frame for a CD.

The CD frame begins with a synchronization header **40**, which is a special sequence of transitions used by the player to detect the beginning of the frame, to calibrate its timing, and to adjust the rotational speed of the media. Following the header is a special control symbol **41**, which is followed by 12 data symbols **42**, four error correction symbols **44**, 12 more data symbols **46**, and another four error correction symbols **48**, for a total of 33 symbols. Each EFM encoded symbol **43** comprises 14 channel bits, and adjacent symbols are separated by a pattern of 3 merge bits **45**. A group of a specified number of successive frames is referred to as a "sector". On a CD, for example, 98 contiguously recorded frames constitute a sector. Sectors recorded on CD digital optical media track are assigned unique numbers, and because each frame is uniquely identified by its number within a sector, and each symbol is uniquely identified by its number within a frame, it is thus possible to uniquely specify any individual symbol on the disk. The present application utilizes the term "address" to refer to the unique position of a specific symbol on an instance of digital optical media.

Error Correction

Transitions are physically represented by microscopic patterns of pits and lands, and excessive physical damage to the media surface, such as a scratch, can obscure the precise location of the region boundaries and thereby corrupt the data reading. To protect against such hazards, the media is written with additional, redundant data in the form of error correcting symbols, also known

as "redundancy symbols". These are mathematically determined to correspond to the other data written on the media in such a way that the player can use them as it reads the media not only to determine if errors have occurred, but also under certain conditions to correct errors. A fixed number of data symbols, together with fixed number of assigned redundancy symbols computed for them form a data structure known as an error correction "codeword", and a symbol can be part of one or more different codewords. Since the redundancy symbols are computed in each case for a specific series of data symbols in the codeword, a change in those data symbols (such as through corruption or damage to the media) may be detected by mathematical operations involving the redundancy symbols. Methods for choosing and implementing appropriate error correction coding are well-known in the art. Many of these methods utilize implementations of the Reed-Solomon error-correcting algorithm. For example, the error correction code (ECC) employed in the CD system is referred to as the "Cross-Interleave Reed-Solomon Code" (CIRC). There is an additional ECC level used in CD-ROM's known as the "Reed-Solomon Product-Like Code" (RSPC). This RSPC error-correcting system is also used in DVD's.

Invalid Symbols

The present application utilizes the term "erroneous symbol" to refer to any symbol which will cause the error-correcting system of a digital optical media player to detect that a symbol in a particular codeword does not have the proper value. The present application utilizes the term "invalid symbol" to refer to any channel bit sequence which can be written to digital optical media but which does not correspond to a possible symbol value as specified by the bi-directional mapping cited in the applicable standards for the digital optical media. By its very nature, an invalid symbol will always be detected by the media player as an erroneous symbol. Moreover, it will be detected as an erroneous symbol immediately by the player's decoder, even before the application of any error-detecting and correcting system.

As the optical media player reads the channel bit sequences corresponding to each symbol from the digital optical media, it checks them to make sure that they are valid. For example, if a CD player encounters a channel sequence with two transitions less than 3 or more than 11 time units apart, it flags that symbol as invalid, since the channel bit sequences used in CD are the EFM code sequences and no valid EFM code sequence has such transitions. An error of this sort is referred to as a "run length limited" (RLL) error.

There are also certain symbol patterns which are not used in the EFM bi-directional mapping of the 256 byte values. Out of a total of 267 different EFM symbols that obey the RLL rules, 256 are used to represent the different values of data bytes which are possible, and two are reserved for special use, leaving nine symbols which obey the RLL rules but which do not appear in the EFM bi-directional mapping and are therefore undefined. These undefined symbols are also detected by the EFM decoder as invalid codes. The present application will use the term "undefined symbol" to refer to such a symbol whose channel bit code sequence is unused and undefined by the bi-directional mapping of applicable standards.

Whether an invalid symbol causes an RLL error, or is simply an undefined symbol, the invalid symbol is flagged by the decoder. A flagged symbol represents an error in a known position and is referred to as an "erasure". Under normal circumstances an invalid symbol implies that the media has been damaged in such a way that the transition timing cannot be read properly. In certain conditions, such errors may be corrected by the ECC decoder.

It is also possible that a symbol might have been damaged, but in such a way that it corresponds to a valid channel bit sequence. In this case, there is an error, but it cannot be immediately recognized as such, nor can its position be immediately detected. To guard against such errors, the ECC decoder checks every recovered ECC codeword to determine if any of its constituent symbols have been read erroneously. In certain conditions, such errors may also be corrected by the ECC decoder, but the number of such errors which are

correctable is only half as large as the number of correctable erasures (since the position of an erasure is known when it is detected, but the position of a non-erasure error must be separately determined).

Data Interleaving

Symbols occurring sequentially on the data track of digital optical media do not correspond to sequential bytes of in the input stream of user data. Rather, each byte of sequential data is assigned a non-sequential address within a specific frame when its corresponding symbol is written onto the disk. This is known as "interleaving", and is schematically illustrated in Fig. 5, to which reference is now briefly made. A series of sequential data bytes **50** divided into groups **52** of 24 bytes is mapped into a corresponding series of symbols which are sequentially recorded on the data track of the CD. As previously mentioned, data sequentially recorded on the data track in sequentially divided into frames **56**. A particular data byte **58** is mapped to a symbol in a specific location in a specific frame **60**, and the next byte **62** within the same group is mapped to a symbol in a much later frame **64**. This process is repeated so that the next sequential byte **66** is mapped to an even later frame **68**. In a similar way, previous data byte **70** has been mapped to frame location **72** in the space between later consecutive bytes **58** and **62**. The precise mapping for CD's is illustrated in the table in Fig. 6. In general, every kind of digital optical media will have a mapping between the position of the data bytes which exist in the output stream of a player of the digital optical media and the address of the symbols on the digital optical media which correspond to those data bytes.

The purpose of interleaving is to spread out the physical locations of the sequential input data over the media so that localized damage to the media surface will not grossly impact any one segment of the data. Instead, the effect of the damage will be distributed over a large area. In particular, the interleaving is chosen to sufficiently insure that reasonable levels of digital storage medium imperfection damage will corrupt only a few constituent symbols of an ECC codeword. This will insure that the position of the erroneous symbols in the

codeword can be located and that then correct values can be calculated when the ECC codeword is decoded in such a way that only small number of erroneously recovered data symbols (*i.e.*, errors) will occur in any one contiguous segment of the data.

The input bytes of data to be recorded on a CD are considered to be in groups of 24. The size of this group corresponds to the number of input data symbols within a frame on the media data track. Fig. 6, to which reference is now briefly made, illustrates precisely how sequential bytes of data in 24-byte group *n* (column 74) are mapped to non-sequential frames (column 76) and symbol locations (column 78) within those frames on a CD data track. It may be noted from Fig. 6 that no data is mapped to frame symbol number 0, since this location is where the control symbol is placed. Furthermore, no data is mapped to frame symbol numbers 13, 14, 15, and 16, nor to frame symbol numbers 29, 30, 31, and 32, since these locations are used for the error-correcting redundancy symbols.

The Phases of Data Re-ordering and Error Correction

During the reading operation, the player re-orders the data symbols contained in the sequentially recovered frames recorded on the media into their correct sequence. For CD digital optical media there are two separate phases of the data reordering, and at each phase a different set of error correcting code words are formed and checked (by the ECC decoder) for errors. Other embodiments may employ additional phases. This is schematically illustrated for CD's in Fig. 7, reference to which is now briefly made. The decoded channel bit symbols **80** corresponding to the last 32 symbols of each recovered frame pass through a partial de-interleaving step **82**, comprising a number of delays **81** for selected symbol positions. Then "C1", the first level of error correction **84** is applied to the resulting 32-symbol C1 ECC codeword. This is followed by a further partial de-interleaving step **86**, after which "C2", the second level of error correction **88** is applied to the resulting 28-symbol C2 codeword. The final de-interleaving step **90** results in the data bytes **92** being reassembled into their original input sequence order with the two levels of error correction having been

applied in the process. (The four C1 code redundancy symbols and the four C2 redundancy symbols are discarded after C1 and C2 decoding, respectively.)

In the first phase **84** the C1 decoder may be able to correct errors not only in the 24 data symbols, but also errors in the four C2 redundancy symbols that comprise the C1 codeword. Each of the 24 data symbols in a particular C1 codeword was originally contained in one of 24 different input data frames. Each of the four C2 redundancy symbols belong to one of four different C2 codewords; all four C1 redundancy symbols below to the particular C1 codeword. The C2 codeword obtained in the second level error correction comprises 24 input data symbols taken from only two input data frames and the four C2 parity symbols all belong to the particular C2 codeword.

Each phase of the CIRC error-correcting decoding is able to detect the existence of erroneous symbols in a specific codeword, and within any such codeword may correct up to two erroneous symbols whose locations within the codeword are unknown. Error locations are normally unknown when the erroneous symbols were obtained from valid EFM code sequences, for such errors are not immediately recognizable by the EFM decoding process as such. If, on the other hand, the locations of the erroneous symbols are known, the CIRC method may correct up to four erroneous symbols. Error locations are normally known when the errors correspond to invalid EFM code sequences which were previously marked by the EFM decoder as erasures. Thus, each phase of error correction may correct up to t_e valid symbols which are erroneous, and up to t_r erasures, such that t_e and t_r jointly satisfy the inequality $2t_e + t_r < 5$. Each phase of error correction may thus also correct one valid symbol which is erroneous and two erasures. Errors in excess of these limits will be detected with finite probability, but the player will not be able to correct them. Note that the probability of detecting that an ECC codeword contains an non-correctable number of errors decreases as the number of errors in the codeword increases.

If the decoder fails to detect that a non-correctable number of errors has corrupted a specific codeword, it will deliver (at its option) a valid codeword that is

different from the original (corrupted) codeword. This output codeword will, in general, have symbol values in several codeword locations that are different from those in either the original uncorrupted codeword or the corrupted original codeword.

Thus, the player reorders the data from the symbols recorded on the digital optical media track and while so doing, it attempts to detect errors and to correct them in two related, but distinct, phases. In this manner, under normal conditions a player will be able to recover virtually error-free all the data from digital optical media which has been subjected to ordinary care in handling.

SUMMARY OF THE INVENTION

Optical Media Authentication by Using Special Non-Copyable Symbols

According to the present invention, there is provided digital optical media comprising at least one non-copyable symbol stored thereon. By recording the digital optical media to include such special (non-standard) symbols which can be read by ordinary media readers, but which require special recorders or mastering equipment to write, it will be possible to identify the digital optical media as authentic and distinguish it from an unauthorized copy by detecting such non-copyable symbols. The present application utilizes the term "non-copyable symbol" to refer to a symbol on optical media which is readable as a valid symbol by ordinary players of the optical media and which is distinguishable from a regular symbol by ordinary players of the optical media, but which does not conform to the applicable specifications for the optical media in such a way that it is not reproducible by ordinary recorders and mastering equipment for the optical media. Such special recorders and mastering equipment would not be generally available to either the public or to commercial optical media replicators. When optical media is recorded or manufactured with these non-copyable symbols, an ordinary player will be able to detect them on the original authentic media. Unauthorized copies, however, will not have them, because the ordinary recorders or mastering equipment used to make the copy will be unable to reproduce them. In every other respect, though, the unauthorized copies may be identical to the original.

To verify the authenticity of optical media when this scheme is utilized, the media is scanned by the reader to detect these non-copyable symbols. If they are present, the optical media is determined to be authentic. Otherwise, it is determined to be an unauthorized copy. This determination can be done automatically as the optical media is read (such as by computer) without the need for a separate physical inspection of the optical media.

Copy Protection by Using Patterns of Non-Copyable Symbols

According to the present invention, non-copyable symbols can be used not only for optical media verification, but also for copy protection. There are several ways of implementing copy protection using non-copyable symbols. A simple way is for the software on the optical media to check for the presence of the non-copyable symbols. If they are present, the software will have determined that the optical media is authentic and will execute the application. Otherwise, the software will have determined that the optical media is an unauthorized copy and will refuse to run. This normally renders the copy unusable, but does not provide a high degree of security against schemes which seek to defeat the copy protection. According to the present invention, another method of copy protection which provides a higher degree of security is to encrypt the data on the optical media using an acceptable encryption method, such as the DES algorithm, and to encode the decryption key in a special pattern of non-copyable symbols. The present application will utilize the term "signature" to refer to such a special pattern of non-copyable symbols on optical media, such that information, such as a decryption key, may be encoded within the special pattern. Software on the optical media will recover the decryption key from the signature and employ it to decrypt the data on the media so that it will be usable. If the optical media is an unauthorized copy, however, it will lack the signature, and hence the software will be unable to decrypt the data on the disk. In this way, an unauthorized copy is rendered unusable.

Creating Non-Copyable Symbols

Further according to the present invention, it is possible to create non-copyable symbols for digital optical media by carefully departing from the standards for writing data so that the decoded value of the symbol is no longer well-defined, but such that the reader will interpret the symbol as a valid symbol having one of two different predetermined data values which will be randomly interpreted for the symbol during each separate reading operation. The present application will utilize the term "ambiguous symbol" to refer to such a

non-copyable symbol whose data value is not well-defined according to the standards for digital optical media.

Overriding Error Correction

In ordinary use, error detection and correction is desirable, as it reduces the probability that the data read from digital optical media and delivered to the user will become corrupted by minor physical damage to the media. Error-correcting, however, interferes with the reading of the (non-copyable) ambiguous symbols which serve to identify optical media as authentic, since the ambiguous values will be adjusted by the error correcting circuitry to stable values. It is therefore necessary to override the error correction in specific locations, so that the ambiguous symbols can be detected.

A simple way of bypassing error correction in the writing of ambiguous symbols is to write them in the final sector of the lead-out area of a CD-ROM disk, for example. This is not completely satisfactory, however, as it restricts the location of the ambiguous symbols. The present invention overcomes this disadvantage by providing a method for overriding the error correction in arbitrary locations on digital optical media, thus allowing ambiguous symbols to be written in many locations.

The overriding of error correction according to the present invention makes use of the fact that the mathematical principles of the Reed-Solomon error-correcting algorithm do not distinguish between an error in the data symbols and an error in the error correcting redundancy symbols themselves. The overriding of the ECC is accomplished by causing a non-correctable pattern of erroneous symbols to occur in the ECC portion of the codeword. The non-correctable error pattern is immediately recognized by the ECC decoder as being non-correctable. The ECC decoder will not attempt to change the values of any symbols of an ECC codeword that is corrupted by the detected non-correctable error pattern.

Moreover, the overriding of the ECC is applicable to a wide variety of digital optical media, including CD, CD-ROM, and DVD, and it is usable for digital optical media recordings which are both mass-produced by plastic molding apparatus as well as those which are produced in small quantities by individual recorders controlled by desktop computers, workstations, and the like.

The method of the present invention by which correction of erroneous and/or erased symbols recovered from digital optical media by the media reader's ECC decoder is overridden involves several innovations in writing and formatting data to digital optical media. The present method further discloses a class of special patterns to be recorded onto digital optical media, consisting of invalid channel bit sequences written on the digital optical media.

Furthermore, in accordance with an embodiment of the present invention, fourteen bit codes which are invalid are marked onto CDDA, such as compact disks. A standard CD reader will read these invalid fourteen bit codes ambiguously (i.e. sometimes the CD reader will produce one eight bit code and sometimes it will produce a different eight bit code for the same invalid fourteen bit code).

Additionally, the present invention utilizes the invalid codes for at least two different purposes. In one embodiment, the present invention utilizes the invalid codes to mark compact disks with identifying data. Thus, specific invalid fourteen bit codes are written to specific locations on the compact disk. The present invention may incorporate an identifying unit which checks the compact disk and identifies which, if any, of the fourteen bit codes thereon are invalid. The checking operation utilizes the fact that the invalid data will provide ambiguous data when read. Thus, the checking operation repeatedly reads the locations where markings are expected. If the data in the specified locations is not identical for each reading (i.e. sometimes it has one value and sometimes it has another value), then the compact disk is marked with invalid codes. The identifying unit can also determine if the invalid fourteen bit codes correspond, in location and code values, to codes which identify the compact disk as being marked.

In a further embodiment, the present invention utilizes the marked compact disk as a key for protecting software. Before a protected software product performs certain operations, it ascertains that a marked compact disk is present in its compact disk drive. The marked compact disk has to have the markings specific to the protected software. If the invalid codes are in the places specific to the protected software, the software can operate. If the invalid codes are in the wrong places or if there are no invalid codes, the software cannot operate. Furthermore, the key can also function as the encryption and/or decryption key if the data, or a software program, has been encrypted. Upon reading the key, decryption software can decrypt the data.

Finally, a system is disclosed for recording invalid symbols, on digital optical media mastering and recording apparatus.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention is herein described, by way of example only, with reference to the accompanying drawings, wherein:

Fig. 1 is a schematic illustration of a data-storage layer of prior art digital optical media;

Fig. 2 is a partial listing of prior art EFM codes used to map 8-bit byte values to constrained 14-bit channel sequences;

Fig. 3 is a schematic illustration of digital signals detected from prior art digital optical media;

Fig. 4 is a schematic illustration of a frame of data on prior art digital optical media;

Fig. 5 is a conceptual illustration of data interleaving on prior art digital optical media;

Fig. 6 is a table showing the prior art mapping of input (user) data bytes to frame and symbol locations on a digital optical media track;

Fig. 7 is a schematic illustration of the symbol reading, error checking, and reordering operations of a prior art digital optical media player;

Fig. 8 is a table showing unused EFM channel bit sequences that are usable for invalid symbol representation;

Fig. 9 illustrates some invalid EFM channel bit code sequences with run-length-limited (RLL) errors;

Fig. 10A is a table showing the mapping of data bytes to their corresponding frame locations and certain C1 redundancy symbols on an audio CD;

Fig. 10B is a table showing the mapping of data bytes to their corresponding frame locations and certain C1 redundancy symbols on a CD-ROM;

Fig. 11 is a table showing the mapping of data bytes to their corresponding C2 redundancy symbols and related C1 redundancy symbols;

Fig. 12 is a schematic block diagram illustration of a mechanism for recording non-copyable symbols onto digital optical media;

Fig. 13 is a schematic illustration of an ambiguous symbol recorded onto an original CD, in accordance with a preferred embodiment of the invention and the alternative interpretations of the symbol by a digital optical media player;

Fig. 14 is a schematic illustrations of an ambiguous symbol recorded onto an original CD, in accordance with a further preferred embodiment of the invention and the alternative interpretations of the symbol by a digital optical media player;

Fig. 15 is a schematic illustration of a marked compact disk, constructed in accordance with a preferred embodiment of the present invention and two compact disks copied therefrom;

Fig. 16 is a block diagram illustration of a unit for creating the marked compact disk of Fig. 15;

Fig. 17 is a flow chart illustration of a method of identifying the marked compact disk of Fig. 15;

Fig. 18 is a block diagram of a system for encrypting software utilizing the marked compact disk of Fig. 15; and

Fig. 19 is a block diagram of a system for decrypting software utilizing the marked compact disk of Fig. 15;

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention describes a method for identifying original digital optical media and distinguishing original optical media from unauthorized copies by recording non-copyable ambiguous symbols on the media and protecting them by overriding the error correction which is normally in effect. The present invention furthermore describes a method for copy protection of digital optical media. The present invention moreover describes a method of overriding the error correction normally in effect by replacing the error correction codes (ECC) or symbols with invalid symbols, thereby protecting the ambiguous symbols from being corrected during the media player's attempt to correct what it assumes is an "error".

Creating and Detecting Ambiguous Symbols

In the preferred method for creating ambiguous symbols, one of the transitions is shifted so that it will no longer be synchronized with the channel data detection clock signal. The result is that the decoding of the channel bit code sequence (the symbol) into an 8-bit data value (a byte) is no longer well-defined.

Fig. 13, to which reference is now made, illustrates a symbol, referenced **100** which is recorded onto the authentic original CD, in accordance with a preferred embodiment of the invention. Symbol 100 comprises an ambiguous transition 102 which has been shifted relative to the clock position **103**. The result is that the detected transitions of the EFM code sequence, generally referenced 101 are no longer well-defined. The ill-defined transitions of the EFM code sequence 101 are indicated by "?". The media player will "digitize" the transitions by synchronizing them with the clock positions. Each transition of a code sequence is normally associated with a clock position so that the code sequence represented by a symbol will be well defined. Since, there is no well-defined clock position for shifted transition **102**, there are two possible equally valid digitizations for this transition, referenced 105 and 107, respectively.

If the media player digitizes transition **102** to be at clock position **106**, it will interpret the code sequence as symbol **105**, which decodes to a data byte

with a hexadecimal value of E4. If, on the other hand, the media player digitizes transition **102** to be at clock position **108**, it will interpret the code sequence as symbol **107**, which decodes to a data byte with a hexadecimal value of F8. Subsequent readings of the symbol will result in a random distribution of interpreted data values. Some will be interpreted as E4, while others will be interpreted as F8.

Thus, in order to detect if a symbol is ambiguous, the media player reads the symbol a plurality of times. If the interpreted value shows any variation from one reading to the next, owing to the two possible values for the symbol, then it has detected the presence of an ambiguous symbol. When an ordinary copy is made of the optical media containing one or more ambiguous symbols, the copy will not contain ambiguous symbols, because standard optical media equipment (described hereinabove with respect to Fig. 1) cannot copy the ambiguous symbols Specialized equipment is needed to write ambiguous symbols. Thus, a copy made according to digital optical media technology standards will only contain regular symbols whose values are not ambiguous and which will be read and decoded consistently as having the same value. It will thus be appreciated that, since ambiguous symbols are non-copyable symbols, they can be used for authentication and for copy protection of digital optical media.

Because the detection of ambiguous symbols involves a random process, it is preferable to include a number of ambiguous symbols in order to ensure that at least one of the ambiguous symbols is detected. If enough ambiguous symbols are recorded on an instance of digital optical media, and the media is read a sufficient number of times, the probability of detecting at least one of the ambiguous symbols which was written to the original media approaches unity. Consequently, if an ambiguous symbol is not detected, it is nearly certain that the optical media is an unauthorized copy.

Reference is now made to Fig. 14, which illustrates an alternative embodiment of the invention in which the ambiguous symbol 110 comprises a

smooth transition, referenced 112, between the pits and lands, in contrast to the sharp transition required by the standards.

Since, there is no well-defined clock position for shifted transition **110**, there are two possible equally valid digitizations for this transition, referenced 114 and 116, respectively.

The media player can digitize transition **112** to be at clock position **115**, and thus will interpret the code sequence as symbol **124**, which decodes to a data byte with a hexadecimal value of E4. Alternatively, the media player can digitize transition **112** to be at clock position **117**, which decodes to a data byte with a hexadecimal value of F8. Subsequent readings of the symbol will result in a random distribution of interpreted data values, either E4 or F8.

Overriding Error Correction

As previously described hereinabove, it is necessary to protect symbols from being altered by the media player's error-correcting system. It is the purpose of the error-correcting system to detect, and if possible, to correct any data which is not read precisely as it was originally written according to the applicable standards of the digital optical media. Since ambiguous symbols may be read as having two different data values, at least one of these values will be interpreted as erroneous by the player. But, in order to detect an ambiguous symbol, it is necessary to read different values for different reading operations, so if the value of an ambiguous symbol is altered in an attempt to correct what the reader considers to be an error, it will not be possible to detect the symbol as ambiguous. It is therefore necessary to override the error correction for the ambiguous symbol. It is also necessary, though, not to disable the error correction entirely, as this would lead to excessive reading errors and unacceptable performance. It is therefore necessary to override only the error correction for specific ambiguous symbols. According to the present invention. error correction for an ambiguous symbol may be overridden by intentionally introducing certain errors in the codeword where the ambiguous symbol is located. By so doing, the

error-correcting capabilities of the player will be used up on these deliberate errors and there will be no alteration of the ambiguous symbol.

The description that follows is illustrated by way of example for CDDA media, but it will be appreciated by persons skilled in the art that the principles herein are applicable to digital optical media in general, and that the present invention is not limited to application for that specific embodiment of digital optical media, but will apply to other embodiments as well, including, but not limited to CD-ROM and DVD.

Overriding of error correction affects a particular codeword as a whole, and therefore every symbol within that codeword. It relies on the characteristics of the Reed-Solomon error correction algorithm and is equally applicable to all implementations of that algorithm, including CIRC and RSPC implementations. For the Reed-Solomon algorithm, an error pattern that is composed of t_e errors together with t_x erasures is non-correctable if $2t_e + t_x \geq d_{min}$, where d_{min} is the minimum distance of the ECC ($d_{min} = 5$ for both the C1 and C2 ECC employed in the CD system). Most detectable/non-correctable error patterns will satisfy $d_{min} \leq 2t_e + t_x \leq d_{min} + 1$. In general, to override error correction it is possible to replace a certain number of the symbols in the codeword containing the ambiguous symbol with erroneous symbols.

In an embodiment for a CDDA, such as a CD-ROM, an erroneous symbol can be made to occur at a specific location in a selected ECC codeword by causing an altered 14-bit EFM code sequence to be written to the media track location corresponding to the targeted symbol (other sequence lengths are applicable to different media, such as DVD). Such an erroneous symbol can be any of the following:

(i) a valid but incorrect symbol (*i.e.*, one that corresponds to one of the 255 values that are different from the correct symbol value);

(ii) an invalid symbol that does not contain an RLL error but which does not correspond to any of the possible 256 symbol values or the two special symbol values;

(iii) an invalid symbol which contains an RLL error.

The advantage of using an invalid symbol is that it will be immediately detected as erroneous and hence it will be flagged as an erasure. The advantage of using an invalid symbol which does not contain RLL errors is that it does not exceed any of the timing limits for EFM encoding. This is a preferred embodiment of invalid symbols for CD's.

In a preferred embodiment of overriding error correction, erroneous symbols are substituted within certain ECC codewords at particular locations to create a non-correctable error pattern. Though the digital optical reader of the ECC decoder reading the digital optical media will be able to detect the non-correctable error pattern, it will not attempt to carry out any error correction of the corrupted ECC codeword containing the erroneous and erased symbols. The values of the erroneous and erased symbols will remain unaltered.

In a further preferred embodiment of the invention, the detected non-correctable error pattern comprises errors or erasures principally located in the redundancy bytes of the targeted ECC codewords and further comprises errors or erasures in the input data symbols contained in the targeted codewords.

Although the redundancy symbols will not be returned by the digital optical reader to the requesting software application, the latter altered input data byte will be visible to the application and thus can be used to cause some specific action to occur.

Reference is now made to Figs. 8, 9, 10A and 11 in order to illustrate the method for overriding the error correction codes (ECC) of a specific erroneous or erased symbol on an audio CD.

Fig. 8 illustrates a table of unused 14-bit EFM channel bit sequences (*i.e.*, those sequences are not used to represent any of the possible 256 byte

values in the CD system specification). Each code, represented by a row, referenced r1, r2...r9, does not correspond to a valid channel bit sequence. For example, the channel bit sequence 01001000000000 (row r7) is invalid since it does not correspond to any byte value or other assigned value.

Fig. 9 illustrates two examples of 14-bit channel data sequences, referenced **140** and **142**, which violate the EFM run-length-limited (RLL) channel bit encoding rules. Sequence **140** is invalid because it causes two transitions **142** which are less than three (3) clock periods apart. Sequence **144** is invalid because it causes adjacent transitions **146** which are more than eleven (11) clock periods apart.

It will be appreciated by persons knowledgeable in the art that any other invalid EFM sequences which do not conform to the CD standards may be used. It will also be appreciated by persons knowledgeable in the art that other optical media may use other sequence lengths, such as a 16-bit sequence in DVD systems.

Fig. 10A is a table showing the mapping of 24 sequentially input audio CD sector data bytes to their corresponding locations in the frames that are contiguously recorded along the disk data track. Fig. 11 is a table showing the mapping of data bytes to their corresponding C2 redundancy symbols and the related C1 redundancy symbols on both an audio CD and a CD-ROM.

The table of Fig. 10A comprises five columns, referenced **120**, **122**, **124**, **126** and **114**. Column **120** lists the numbers of the data bytes in the *n*th **24**-byte sector sub-block to be made ambiguous. Columns **122** and **124** list the frames and symbol numbers, respectively, corresponding to the data bytes in column **120**. Columns **126** and **128** list the corresponding C1 error correction symbols in the corresponding frame for specific symbols numbers 29 and 31 (column **126**) and symbols numbers 30 and 32 (column **128**). Specific symbols numbers 29, 30, 31 and 32 are selected by reference to the International Electrotechnique Commission (CEI-IEC) standard 908 ("Red Book") specifications, using analysis criteria, known in the art.

The table of Fig. 11 shows the mapping of data bytes to their corresponding C2 redundancy symbols and the related C1 error correcting symbols (*i.e.*, the C1 redundancy symbols used to correct errors in the C2 redundancy symbols). Fig. 11 comprises four columns, referenced **132**, **134**, **136** and **138**. Column **132** lists the C2 byte and symbol number, column **134** lists the C2 symbol and frame numbers, column **136** lists the frame corresponding to symbols numbers 29 and 31, and column **138** lists the frame corresponding to symbols numbers 30 and 32.

It will be appreciated by persons knowledgeable in the art that the specific steps for overriding error correction will depend on the media and the recorder utilized. For example, CD-ROM recording involves swapping even-numbered and odd-numbered input data bytes, whereas there is no such swapping in audio CD recording. Fig. 10A illustrates the mapping of data bytes to their corresponding frame locations and certain C1 redundancy symbols on an audio CD. Fig. 10B is similar to Fig. 10A and illustrates the mapping of data bytes to their corresponding frame locations and certain C1 redundancy symbols on a CD-ROM.

It will also be appreciated by persons knowledgeable in the art that recorders may "offset" input data by multiples of four bytes and that first logical byte may be located in one of six positions (0, 4, 8, 12, 16, or 20, defined in the Yellow book) in the 24-byte sub-block. These six locations are known as the "minor offset".

The method of overriding the error correction of a specific symbol is described, for the purposes of example only, with reference to an audio CD (Figs. 10A and 11). In this example, it is assumed that the CD recorder places the first logical byte in the first byte sector sub-block, that is with a minor offset of 0.

The method for overriding the error-correction in an audio CD comprises the following steps:

1. Select a data byte, whose error-correction is to be overridden. Identify the data byte by its byte number, and its data sub-block (*n*). For

example, referring to the table of Fig. 6, select byte 9 from data sub-block 86. That is, $n = 86$.

2. Divide the byte number by 4 and examine the quotient. If the quotient is even, replace n with $n + 2$. For example, byte 9 will yield a quotient of 2. Since the quotient is an even number, replace $n = 86$ with $n = 88$.

3. The "C1" level of error correction is first overridden. Reference is now made to Fig. 10 in order to find the frame number and symbol number of the corresponding symbol whose error-correction is to be overridden, as it is located on the digital optical media. Reference is made to the row (referenced **130**) for data byte 9. For n =88, column **126** shows that symbols numbered 29 and 31 need to be replaced in frame 101 (n +13) with invalid symbols. Similarly, symbols numbered 30 and 32 need to be replaced in frame 101 (n +12) with invalid symbols.

4. The second "C2" level of error correction is now overridden. Reference to Fig. 11 shows that it is necessary to replace a total of 20 additional symbols (detailed below) with invalid symbols, and it lists which symbols these are. For byte 9 of data group 86, corresponding to n = 88, reference to column **134**, symbol 13 of frame 137 (n+49), symbol 14 of frame 140 (n+52), symbol 15 of frame 145 (n+57), and symbol 16 of frame 148 (n+60) need to be replaced. With reference to column **136**, symbols 29 and 31 of frames 137, 141, 145 and 149 need to be replaced. Similarly, with reference to column **138**, symbols 30 and 32 of frames 136, 140, 144, and 148 need to be replaced.

Data byte 9 of data group 86, will thus have its error correction overridden. This symbol's value will not be adjusted by the reader's error-correction system and will be read as raw data off the digital optical media.

By following the hereinabove described method, other data bytes of other sub-blocks may be similarly overridden.

Similarly, by reference to Fig. 10B, instead of Fig. 10A, the error-correction of a CD-ROM (with a minor offset of 0) can be overridden.

CD-ROM players utilize a third level of correction, known as the Reed-Solomon Product Code (RSPC) error correction. In a further embodiment of the invention, the RSPC error correction process is initially de-activated, (by means of software, for example) while the player is carrying out its ECC functions and only activated later.

It will be appreciated by persons knowledgeable in the art that recorders should be able to carry out the RSPC correction and thus, all RSPC codewords should be traced in order to identify any combinations of target and parity bytes which will not be ultimately correctable by the RSPC decoding.

It will be further appreciated by persons skilled in the art that different digital optical media employ different formats and error-correcting protocols, and the present invention, illustrated above in particular for an audio CD, will adapt itself to different embodiments depending upon the specific choice of the digital optical media.

It will be further appreciated by persons skilled in the art that the above described overriding technique is applicable to other digital data employing error correction. The method has applications in evaluating the efficiency of digital optical media decoder mechanisms, quality control of digital optical media production, and the writing of signatures on digital optical media for copy protection purposes.

It will be appreciated that there are numerous other embodiments which may be applied for the overriding or circumventing of the error correction symbols. The number of codewords which are rendered non-correctable can be varied and is not restricted to a specific number. Further, there are numerous possible processes for effecting the non-correctable codewords. In a non-limiting example, non-decodable codewords may be effected by the process of causing erroneous target bytes and by causing certain parity bytes to be erroneous when read by CD readers employing different CIRC block decoders. This is especially suitable for

players which have marginal playback channel bit clocks and would be incapable of reading a CD having too many invalid EFM sequences.

Since it is essential that decoders work correctly and are capable of recognizing that the ECC codewords are not decodable, the embodiment utilized may be varied according to the type of decoder and the maximum number of error/erasures it will attempt to correct. For example, three errors per targeted codeword may be created to override the error correction. These errors may use legal (but incorrect) EFM sequences. Alternatively, targeted codewords may affected with five invalid sequences.

It will be further appreciated, by persons skilled in the art, that the location of the target bytes in a sector is not restricted to a particular sector. One non-limiting example of suitable target sector bytes are the parity bytes of the RSPC code used in the CD-ROM. In this case, the changed target bytes will only be visible during a raw read and it will not be possible to determine the location of the errors without decoding the RSPC codewords.

Writing Signatures

Once it is possible to write non-copyable symbols it is possible to write a signature, which encodes a value, such as a decryption key. Persons knowledgeable in the art will appreciate that there are many possible ways to encode a signature using non-copyable symbols. One embodiment of a signature is to assign a non-copyable symbol to each binary digit, or bit, of the information which is to be encoded in the signature. Depending on the value of a bit (a "0" or a "1"), the address of its corresponding symbol would be predetermined differently, in such a manner that the occurrence of a non-copyable symbol in a particular address would uniquely determine the value of a specific bit in the signature.

Apparatus for Writing Ambiguous and Erroneous Symbols

Reference is now made to Fig. 12 which is schematic block diagram illustration of a system, generally designated **150**, for writing ambiguous and

invalid symbols, on digital optical media mastering and recording apparatus, in accordance with an embodiment of the invention.

The system **150** comprises an encoder **162**, laser writing control **164**, and laser **166**, standard components known in the art of digital optical media mastering and recording. Serial data **160** is fed into encoder **162**, which creates a serial data stream of channel bit sequences corresponding to data symbols and the error-correcting symbols, and performs the data interleaving as needed before sending the information in the form of channel bits to laser writing control **164**. After this point, laser **166** writes the channel bits onto the digital optical media or the master for creating the digital optical media.

The system **150** further comprises additional apparatus, generally designated **152**, which comprises a multiplexer **170**, a control unit **172**, an encoder **162**, and ambiguous/invalid symbol generator **174**. The ambiguous/invalid symbol generator creates a serial data stream of channel bit sequences corresponding to ambiguous and invalid symbols. Multiplexer **170** is driven by timing and control unit **172**, which selects its input from encoder **162** and ambiguous/invalid symbol generator **174**. The additional apparatus **152** is inserted between encoder **162** and laser writing control **164**. The direct connection between encoder **162** and laser writing control **164** is broken as shown by symbol X (**168**).

In this embodiment, most of the channel bit data from encoder **162** is passed along by multiplexer **170**, but timing and control unit **172** keeps track of the current frame and symbol that is currently being written, and is programmable to intercept certain symbols selected according to the methods of the present invention. When timing and control unit **172** detects a symbol position that is to be made ambiguous or invalid, it switches the output of multiplexer **170** from the normal channel bit stream from encoder **162**, and substitutes an ambiguous or invalid symbol created by ambiguous/invalid symbol generator **174**, whose timing it also controls.

Thus, the apparatus **150** creates augmented and invalid symbols in predetermined locations on the digital optical media.

Detecting and Reading Ambiguous Symbols

As previously described hereinabove, it is possible to detect ambiguous symbols on digital optical media by comparison of multiple readings of the area of the media on which the ambiguous symbols are located. The media player will read the ambiguous symbols as valid symbols, but will interpret them as having randomly different values. In contrast, the player will interpret a regular symbol which conforms with the standards as always having the same value. Thus, by reading the symbol a number of times, it is statistically possible to detect an ambiguous symbol. Since the address of a symbol is known upon reading it, the player can also detect and decode a signature recorded onto digital optical media according to the present invention.

The present invention includes a marked compact disk which has invalid fourteen bit codes thereon and the various methods for utilizing such a marked compact disk.

Reference is now briefly made to Fig. 15 which illustrates a compact disk, labeled **250**, and one, very enlarged, fourteen bit code **252**. In addition, Fig. **15** illustrates two possible copies **252** and **253** of compact disk **250** with their fourteen bit codes **255** and **257**. The bits storing a zero are marked with a dot and the bits storing a one are marked with an X. Thus, the fourteen bit code **252** has the bit sequence: 0101000001000. However, since this code has two ones, labeled **254** and **256** in Fig. 15, which have less than two zeros therebetween (in fact, there is only one zero, labeled **258**, between them), fourteen bit code **252** is invalid.

A standard compact disk player will read the fourteen bit code **252** as one of a few possible codes. For example, it might read code **252** as its true bit sequence 0101000001000 or it might have read it as bit sequence 0100000001000 or as bit sequence 0010000001000. The compact disk player

will then convert the fourteen bit code it has read to the closest fourteen bit code for which there is an associated eight bit code. For code **252** the two possible close fourteen bit codes are:

255: 01001000010000 corresponding to the eight bit ode for "0"; and

257: 00010000010000 corresponding to the eight bit code for "19".

It is noted that code **255** differs from code **252** in the fourth and fifth bits and code **257** differs from code **252** in the second bit only.

A standard CD player will select one of codes **255** and **257** randomly. Thus, an attempt to copy the compact disk **252** will produce one of compact disks **251** and **253** having codes **255** and **257**, respectively, thereon. Compact disks **251** and **253** are not perfect copies of marked compact disk **250** since a) they have incorrect codes for code **252** and b) since the CD player will always read the same value when reading either of codes **251** and **253** and when reading code **252** will read either code **251** or **253**.

In other words, invalid code **252** provides an ambiguous output when read by a standard compact disk player and therefore, can be utilized as a marking signature on compact disk **250**. Since there are 2^{14} possible fourteen bit codes of which only **267** are valid, there are plenty of invalid codes available for use as identifiers.

There are numerous ways to produce invalid codes on a compact disk. For example, as described hereinabove with respect to Figs. 13A-C and 14A-C, the transition can be shifted relative to the clock position or the code can comprise a smooth transition between the pits and lands, in contrast to the sharp transition required by the standards.

Alternatively, the pits and/or lands can be shorter and/or longer than the standard. Thus, a pit might represent less than three or more than eleven bits.

Fig. 16, to which reference is now made, illustrates one add-on system **278** to be added to a bitstream generator **280** of a standard compact disk writer, such as writer **46**, for creating the non-standard surface **60** of Fig. 15. The bit stream generator **280** receives the actual data to be stored and format data (indicating the location at which a bit is to be written) and creates therefrom a bitstream to be written onto the compact disk. To create the surface **60** of Fig. 15, add-on system **270** modifies the bitstream. The add-on system **278** includes a synchronization generator **282**, an error generator **284**, a summer **286** and an AND gate **288**. Synchronization generator **282** produces a synchronization ("sync") signal at the moment when a bit of data is present. AND gate **288** produces a control signal when the location data (also input to the bitstream generator **280**) coincides with a synchronization signal. Alternatively, AND gate **288** can produce the control signal only when particular location data of interest is received.

The control signal activates the error generator **284** to provide a one bit to the summer **286**. Summer **286** combines the one bit with the bitstream thereby changing the bit of interest to a one. The summer **286** provides the resultant, modified bitstream to a laser modulator forming part of the CD writer.

Reference is now made to Fig. 17 which illustrates one method for identifying marked disks with modified data using a standard CD reader. The method begins by determining, in loop **290**, the locations of the invalid data words on the disk. This is found by indicating (step 292) to the compact disk reader to read the compact disk. If the compact disk reader finds a bad word in a sector, it will consider the sector to be bad. For each bad sector which the compact disk reader finds, the method performs the loop **294**, finding the invalid word (step 296) and entering the invalid word into a table (step 298). The table typically lists the bad sector and the location of the invalid word within the bad sector.

Loop **290** is repeated a number N of times. Once loop **290** has finished, the resultant table is sorted (step 300) according to the bad sectors. In step 302,

the sorted table is reviewed to determine if the invalid words always fall in similar locations (e.g. in the same locations or in their neighboring locations). If this is true for none of the words, then the compact disk is a fraud and step 303 labels it as such. Otherwise, in step 304, the invalid words at the similar locations are checked to see if they have two different values. If so, then the compact disk has ambiguous data. The disk is then a marked disk, as noted in step 305.

If desired, the locations and values of the invalid words at the similar locations (e.g. the "invalid words of interest") can be compared against a signature of marked words to identify the compact disk. Step 306 determines whether or not the signature formed from the invalid words of interest match a desired signature. If so, the compact disk being read is identified as the correct disk. Otherwise, it is a disk marked by the method of the present invention but with the wrong signature.

It will be appreciated that the method of Fig. 17 is operative for invalid fourteen bit codes within the data portion and within the error correction, or other control bit, sections of the frame.

It will further be appreciated that the method of Fig. 17 both identifies that a disk is marked and determines the particular pattern of marks (e.g. the signature) on the marked compact disk. As described in U.S. Patent Application 08/653,205, filed May 24, 1996, entitled "Encryption Key" and which is incorporated herein by reference, the pattern of marks can be utilized as a software protection key. For this embodiment, each software program includes a section which continually or occasionally checks a) that the marked compact disk is present and b) that the pattern of marks thereon matches the signature for the software program. The latter operation is performed by the method of Fig. 17. If conditions a) and b) are met, the software program continues to operate. Otherwise, the software program will stop operating.

Furthermore, as described in U.S. Patent Application **08/653,205**, the key can be utilized as a decryption key, for decrypting some or all of the software to be executed.

Reference is now made to Figs. **18** and **19** which are block diagram illustrations of the system for encrypting and decrypting software, respectively.

The software encryptor **310** comprises the unencrypted software **312**, an encryption key **316** and a data encryption standard (DES) encryptor **314**, such as are known in the art. The software manufacturer uses software encryptor **310**, together with the encryption key **316** supplied by the compact disk manufacturer, in order to encrypt the unencrypted software **312**. The resulting encrypted software **318** is supplied to the customer.

The software decryptor **320** comprises DES decryptor **333**, a marked compact disk **322**, such as any of those described hereinabove, the encrypted software **318** and a key extractor **326**. The key extractor **326**, which utilize the method of Fig. 17, is used to generate the decryption key **330** from the pattern of marks on the marked compact disk **322**. The extracted decrpytion key **330** is then used to decrypt the encrypted software **318** to generate the decrypted software **332**.

A customer wishing to install an application receives a marked compact disk **322**, the encrypted software **318** and installation software **318**. Key extractor **316** is a component of the installation software **328**. To install the software, the customer inserts marked compact disk **322** into a drive of his computer system (not shown). The installation software **328** reads the marked compact disk **322** and extracts a decrpytion key **330** therefrom. The installation software **328** then uses the extracted decrpytion key **330** to decrypt the encrypted software **318** and install the software application.

If a legitimate marked compact disk is used by the customer, then the extracted encryption key **330** matches encryption key **316**, the decrypted software **332** will match the unencrypted software **312**, and the original software program is successfully reconstructed.

If an unmarked or incorrectly marked compact disk is used, the key generated therefrom will not match the original encryption key **316**, and consequently, the decrypted software will not be the same as the original software **312**. Thus, since each installation requires a genuine marked compact disk, a software manufacturer can easily protect his software from unauthorized copying.

It will be appreciated by persons skilled in the art that the present invention is not limited to what has been particularly shown and described above. Rather the scope of the present invention is defined only by the claims which follow.

CLAIMS

1. A digital optical media having at least one non-copyable symbol stored thereon.

2. Digital optical media according to claim 1, wherein said at least one non-copyable symbol has a pre-determined address.

3. Digital optical media according to either of claims 1 or 2 wherein said at least one non-copyable symbol is an ambiguous symbol.

4. Digital optical media according to claim 1, wherein said ambiguous symbol has at least one predetermined data value in accordance with bi-directional mapping.

5. Digital optical media according to any of claims 3 - 4, and comprising at least one codeword, said ambiguous symbol being part of said at least one codeword.

6. Digital optical media according to claim 5, and wherein said at least one codeword comprises at least one erroneous symbol.

7. Digital optical media according to claim 6, wherein said at least one erroneous symbol is an invalid symbol.

8. Digital optical media according to claim 7 and wherein said invalid symbol does not conform with either at least one run length-limited rule and/or bi-directional mapping.

9. Digital optical media according to claim 7, and wherein said at least one codeword comprises at least one assigned redundancy symbol and said invalid symbol corresponds to said at least one assigned redundancy symbol.

10. Digital optical media according to any of claims 3 - 9 and further comprising regions of pits and lands formed therein, said regions specified as having lengths which are integer multiples of an elementary length unit, and wherein the length of at least one region of said

ambiguous symbol is not an integer multiple of said elementary length unit.

11. Digital optical media according to any of claims 3 - 9 and further comprising regions of pits and lands therein, wherein at least two adjacent regions of said ambiguous symbol do not have an edge between them.

12. Digital optical media according to any of the previous claims and comprising any type of optical media including a list comprising Compact Disc Digital Audio (CDDA), Compact Disc Read-Only Memory (CD-ROM), and Digital Video Disc (DVD).

13. A compact disk having at least one invalid fourteen bit code stored thereon in at least one known location of said disk.

14. A compact disk according to claim 13 and wherein said at least one invalid fourteen bit code has a known value.

15. A compact disk according to claim 13 - 14 and comprising a polycarbonate layer having pits and lands therein, said pits and lands generally having widths which are a multiple of a period defined by the frequency of a clock signal, said pits and lands having transitions synchronized with said clock signal, and wherein at least one transition of said at least one invalid fourteen bit code is not synchronized with said clock signal.

16. A compact disk having a multiplicity of words stored therein, wherein at least one of said words produces ambiguous results when read a multiplicity of times.

17. A method for writing a non-copyable symbol on digital optical media, comprising the step of writing an ambiguous symbol on the digital optical media in a predetermined address in a predetermined codeword.

18. A method according to claim 17 and further comprising writing at least one erroneous symbol in said predetermined codeword.

19. A method according to claim 18 and wherein said at least one erroneous symbol is an invalid symbol.

20. A method according to claim 19 and wherein said invalid symbol does not conform with either at least one run length-limited rule and/or bi-directional mapping.

21. A method according to any of claims 19 - 20 and further comprising the step of assigning at least redundancy symbol to said at least one codeword, wherein said invalid symbol corresponds to said at least one assigned redundancy symbol.

22. A method for writing an ambiguous symbol on digital optical media, comprising the step of substituting the channel bit sequence corresponding to the ambiguous symbol for the channel bit sequence corresponding to a symbol originally intended to be written on the digital optical media.

23. A method for writing a signature on digital optical media, comprising the step of writing a plurality of non-copyable symbols on the digital optical media, each of said non-copyable symbols having a predetermined address on the digital optical media selected from a unique set of addresses corresponding to each of said non-copyable symbols, said set containing at least one address.

24. A method for authenticating the validity of a digital optical media comprising the steps of:

a. performing a first reading of a predetermined address of the digital optical media;

b. performing a second reading of said predetermined address of the digital optical media;

c. making a comparison of said first reading with said second reading; and

d. if said first reading is not identical to said second reading, identifying said digital optical media to be authentic.

25. A method according to claim 24 and wherein said a codeword is located at said predetermined address, said codeword comprising an ambiguous symbol.

26. A method for digital optical media copy protection, comprising the steps of:

a. performing an encryption of the data to be recorded onto the digital optical media, said encryption having a decryption key and resulting in encrypted data;

b. providing software which employs said decryption key to perform a decryption of said encrypted data to recover said data to be recorded;

c. encoding said decryption key in a signature; and

d. writing said encrypted data, said software, and said signature onto the digital optical media.

27. A method of protecting compact disks, the method comprising the step of storing data on a compact disk having marked and non-marked areas therein, wherein at least one word from said marked areas produces ambiguous values when read multiple times.

28. A method of selecting among marked and non-marked compact disks, the method comprising the steps of:

a. reading data from a compact disk with a CD reader a multiplicity of times; and

b. determining which words, if any, on said compact disk were read ambiguously by said CD reader and, if so, labeling said compact disk as marked.

29. A method according to claim 28 and also comprising the step of identifying if the marked compact disk has the desired signature of ambiguous words.

30. A method according to claim 29 and wherein said step of identifying includes the step of comparing the ambiguously read words and their locations with said desired signature.

31. A method according to claim 29 and wherein said desired signature forms a key for use by a software protection device.

32. A system for encrypting data comprising:

a. a marked digital optical media having a pattern of marked sectors formed thereon;

b. a key generating unit for generating an encryption key from said marked digital optical media; and

c. an encoding unit for encoding said data in accordance with said encryption key.

33. A system for decrypting encoded data comprising:

a. a key generating unit for generating an encryption key from a pattern of marked sectors on a marked digital optical media; and

b. a decryptor for decrypting said encoded data with said encryption key.

34. A method for encrypting data comprising the steps of:

a. generating an encryption key from a pattern of marked sectors on a marked digital optical media containing a plurality of sectors; and

b. encrypting the data in accordance with said encryption key.

35. A method for decrypting encoded data comprising the steps of:

a. generating an encryption key from a pattern of marked sectors on a marked digital optical media having two sides, each of said two sides containing a plurality of sectors; and

b. decrypting said encoded data with said encryption key.

36. A system for protecting software comprising:

a. a key generating unit for generating an encryption key from a pattern of marked sectors on a marked digital optical media;

b. an encoding unit for encoding said software in accordance with said encryption key;

c. a key extractor for generating a decryption key from said pattern of marked sectors on said digital optical media; and

d. a decryptor for decrypting said software using said decryption key.

37. A method for overriding the error correction in arbitrary locations on digital optical media, comprising the step of writing a non-correctable pattern of erroneous symbols in the error correction code portion of the codeword.

38. A method according to claim 37 wherein the step of writing comprises the step of writing invalid channel bit sequences on said digital optical media.

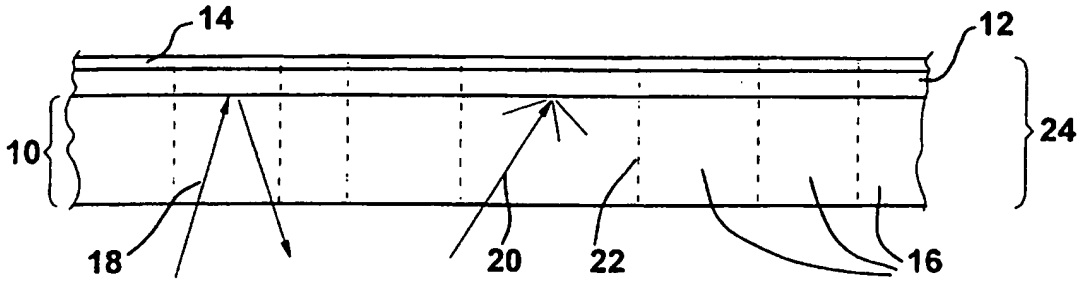


FIG. 1

BYTE VALUE	CHANNEL BIT EFM CODE
00	01001000100000
01	10000100000000
02	10010000100000
03	10001000100000
04	01000100000000
05	00000100010000
06	00010000100000
07	00100100000000

26 28

FIG. 2

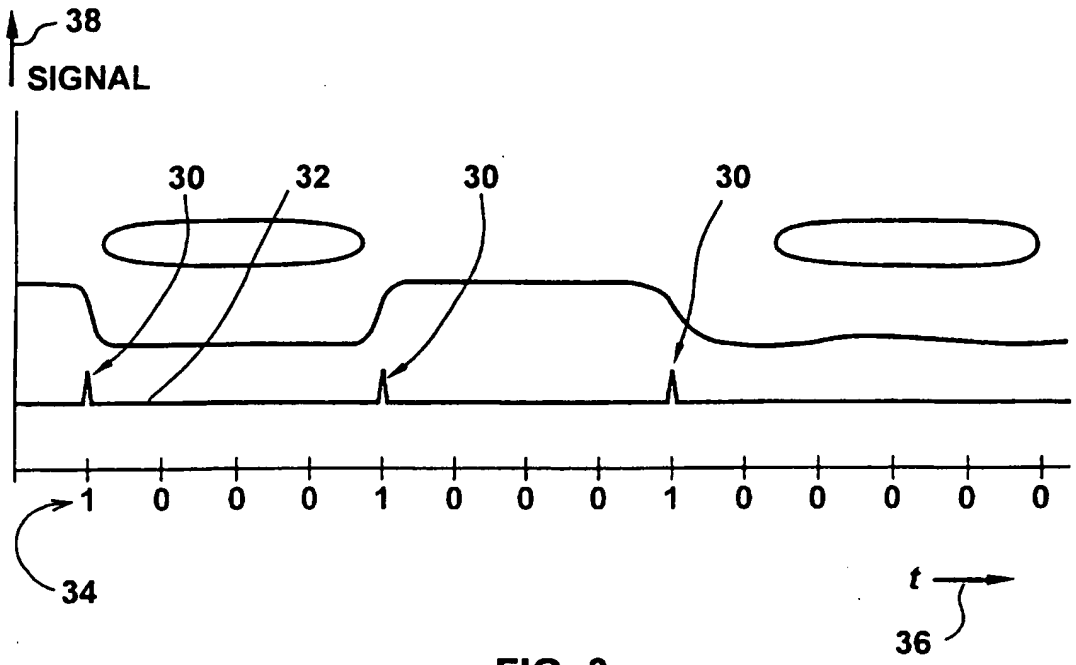

38
SIGNAL
30
32
30
30
1 0 0 0 1 0 0 0 1 0 0 0 0 0
34
t
36

FIG. 3

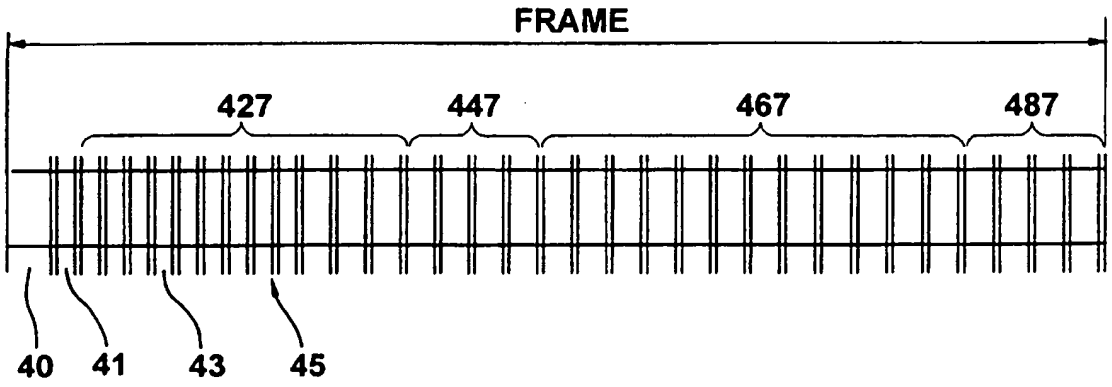

FRAME
427
447
467
487
40
41
43
45

FIG. 4

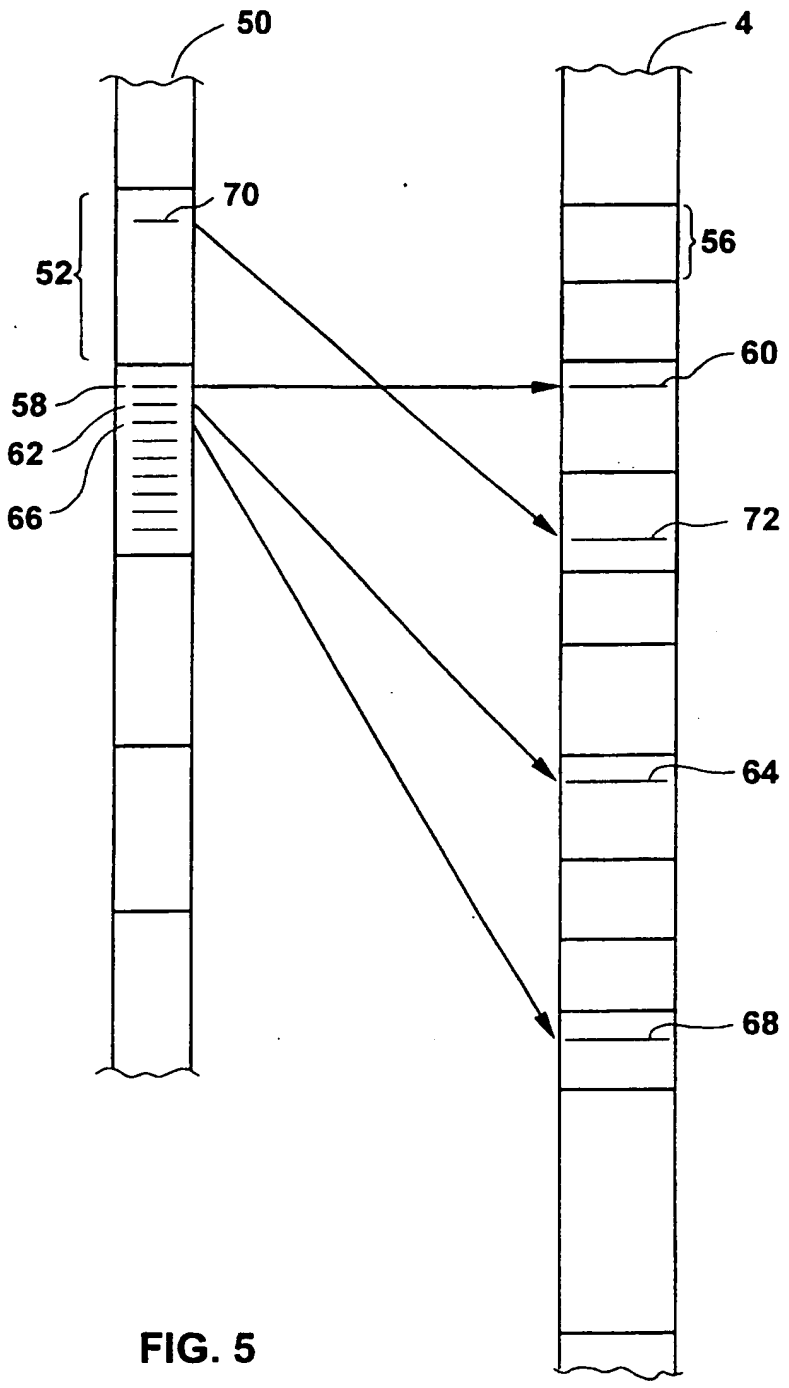


FIG. 5

74	76	78
DATA GROUP *n*	DIGITAL OPTICAL MEDIA TRACK	
BYTE NUMBER	FRAME NUMBER	SYMBOL NUMBER
0	*n* + 3	1
1	*n* + 6	2
2	*n* + 27	7
3	*n* + 30	8
4	*n* + 65	17
5	*n* + 68	18
6	*n* + 89	23
7	*n* + 92	24
8	*n* + 11	3
9	*n* + 14	4
10	*n* + 35	9
11	*n* + 38	10
12	*n* + 73	19
13	*n* + 76	20
14	*n* + 97	25
15	*n* + 100	26
16	*n* + 19	5
17	*n* + 22	6
18	*n* + 43	11
19	*n* + 46	12
20	*n* + 81	21
21	*n* + 84	22
22	*n* + 105	27
23	*n* + 108	28

FIG. 6

FIG. 7

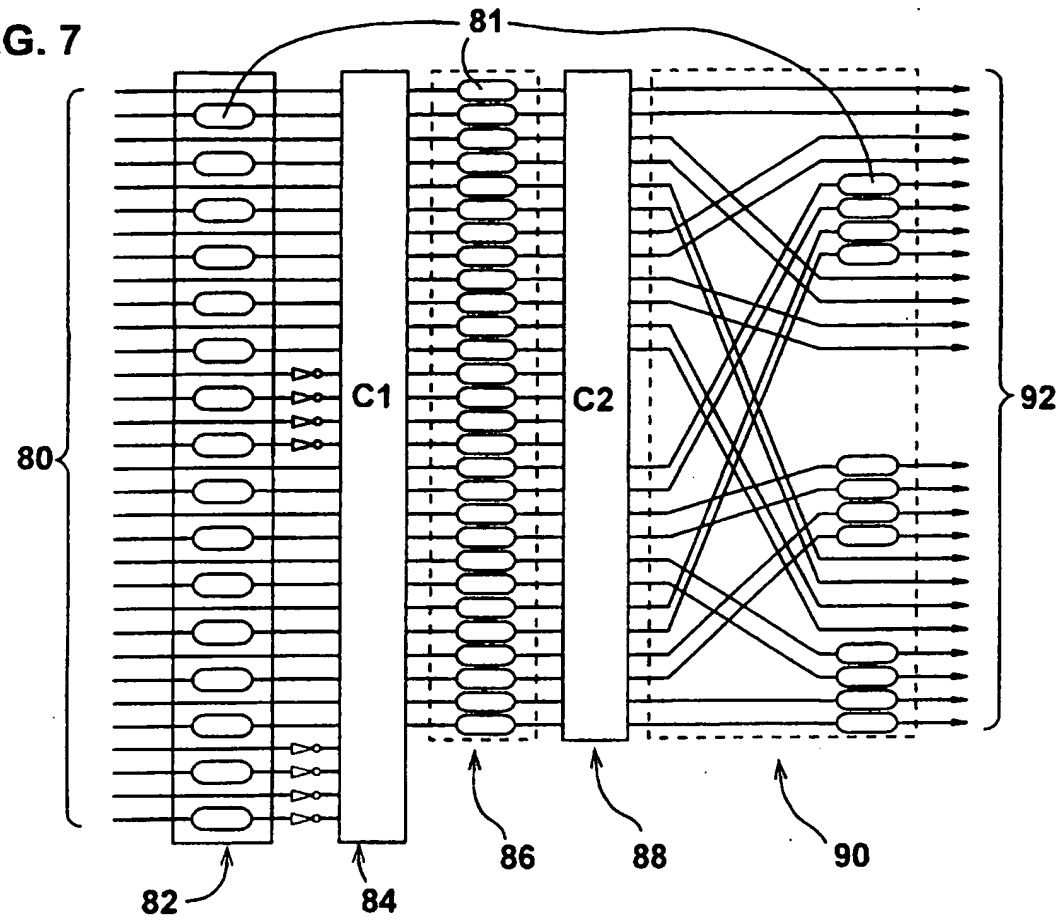


FIG. 8

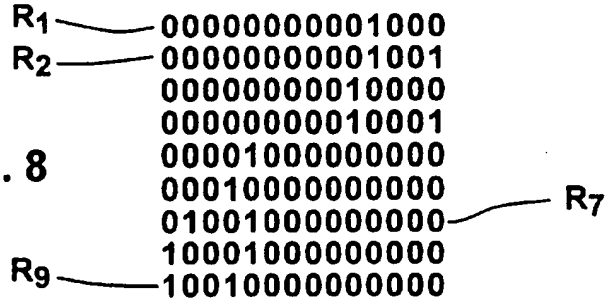
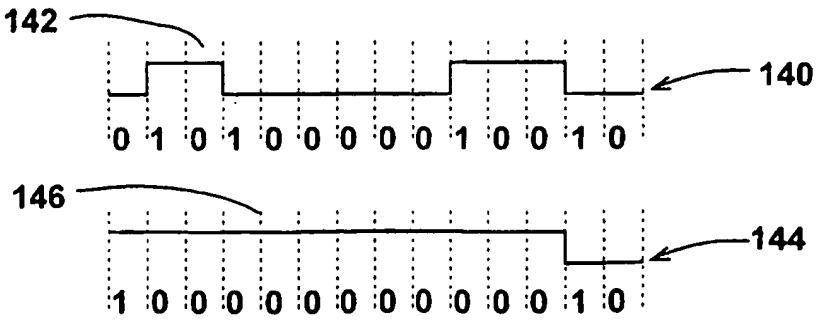


FIG. 9



120	122 / 124		126	128
*n*th 24-byte sector sub-block Data Byte to be made Ambiguous	Symbol to be made Ambiguous		C1 Error-Correction Symbols to be made invalid	
	Frame	Symbol Number	Symbol Numbers 29, 31 in Frame	Symbol Numbers 30, 32 in Frame
0	*n* + 1	1	*n* + 1	*n*
1	*n* + 4	2	*n* + 5	*n* + 4
2	*n* + 25	7	*n* + 25	*n* + 24
3	*n* + 28	8	*n* + 29	*n* + 28
4	*n* + 65	17	*n* + 65	*n* + 64
5	*n* + 68	18	*n* + 69	*n* + 68
6	*n* + 89	23	*n* + 89	*n* + 88
7	*n* + 92	24	*n* + 93	*n* + 92
8	*n* + 9	3	*n* + 9	*n* + 8
9	*n* + 12	4	*n* + 13	*n* + 12
10	*n* + 33	9	*n* + 33	*n* + 32
11	*n* + 36	10	*n* + 37	*n* + 36
12	*n* + 73	19	*n* + 73	*n* + 72
13	*n* + 76	20	*n* + 77	*n* + 76
14	*n* + 97	25	*n* + 97	*n* + 96
15	*n* + 100	26	*n* + 101	*n* + 100
16	*n* + 17	5	*n* + 17	*n* + 16
17	*n* + 20	6	*n* + 21	*n* + 20
18	*n* + 41	11	*n* + 41	*n* + 40
19	*n* + 44	12	*n* + 45	*n* + 44
20	*n* + 81	21	*n* + 81	*n* + 80
21	*n* + 84	22	*n* + 85	*n* + 84
22	*n* + 105	27	*n* + 105	*n* + 104
23	*n* + 108	28	*n* + 109	*n* + 108

FIG. 10A

120 126 128

nth 24-byte sector sub-block Data Byte to be made Ambiguous	Symbol to be made Ambiguous		C1 Error-Correction Symbols to be made invalid	
	Frame	Symbol Number	Symbol Numbers 29, 31 in Frame	Symbol Numbers 30, 32 in Frame
0	*n* + 4	2	*n* + 5	*n* + 4
1	*n* + 1	1	*n* + 1	*n*
2	*n* + 28	8	*n* + 29	*n* + 28
3	*n* + 25	7	*n* + 25	*n* + 24
4	*n* + 68	18	*n* + 69	*n* + 68
5	*n* + 65	17	*n* + 65	*n* + 64
6	*n* + 92	24	*n* + 93	*n* + 92
7	*n* + 89	23	*n* + 89	*n* + 12
8	*n* + 12	4	*n* + 13	*n* + 88
9	*n* + 9	3	*n* + 9	*n* + 8
10	*n* + 36	10	*n* + 37	*n* + 36
11	*n* + 33	9	*n* + 33	*n* + 32
12	*n* + 76	20	*n* + 77	*n* + 76
13	*n* + 73	19	*n* + 73	*n* + 72
14	*n* + 100	26	*n* + 101	*n* + 100
15	*n* + 97	25	*n* + 97	*n* + 96
16	*n* + 20	6	*n* + 21	*n* + 20
17	*n* + 17	5	*n* + 17	*n* + 16
18	*n* + 44	12	*n* + 45	*n* + 44
19	*n* + 41	11	*n* + 41	*n* + 40
20	*n* + 84	22	*n* + 85	*n* + 84
21	*n* + 81	21	*n* + 81	*n* + 80
22	*n* + 108	28	*n* + 109	*n* + 108
23	*n* + 105	27	*n* + 105	*n* + 104

FIG. 10B

FIG. 11

132 C2 Byte, and Symbol Number	134 C2 Symbol Number and Frame Number	136 Symbol Numbers 29, 31 in Frame	138 Symbol Numbers 30, 32 in Frame
$C2_0$	$n + 49, 13$	$n + 49$	$n + 48$
$C2_1$	$n + 52, 14$	$n + 53$	$n + 52$
$C2_2$	$n + 57, 15$	$n + 57$	$n + 56$
$C2_3$	$n + 60, 16$	$n + 61$	$n + 60$

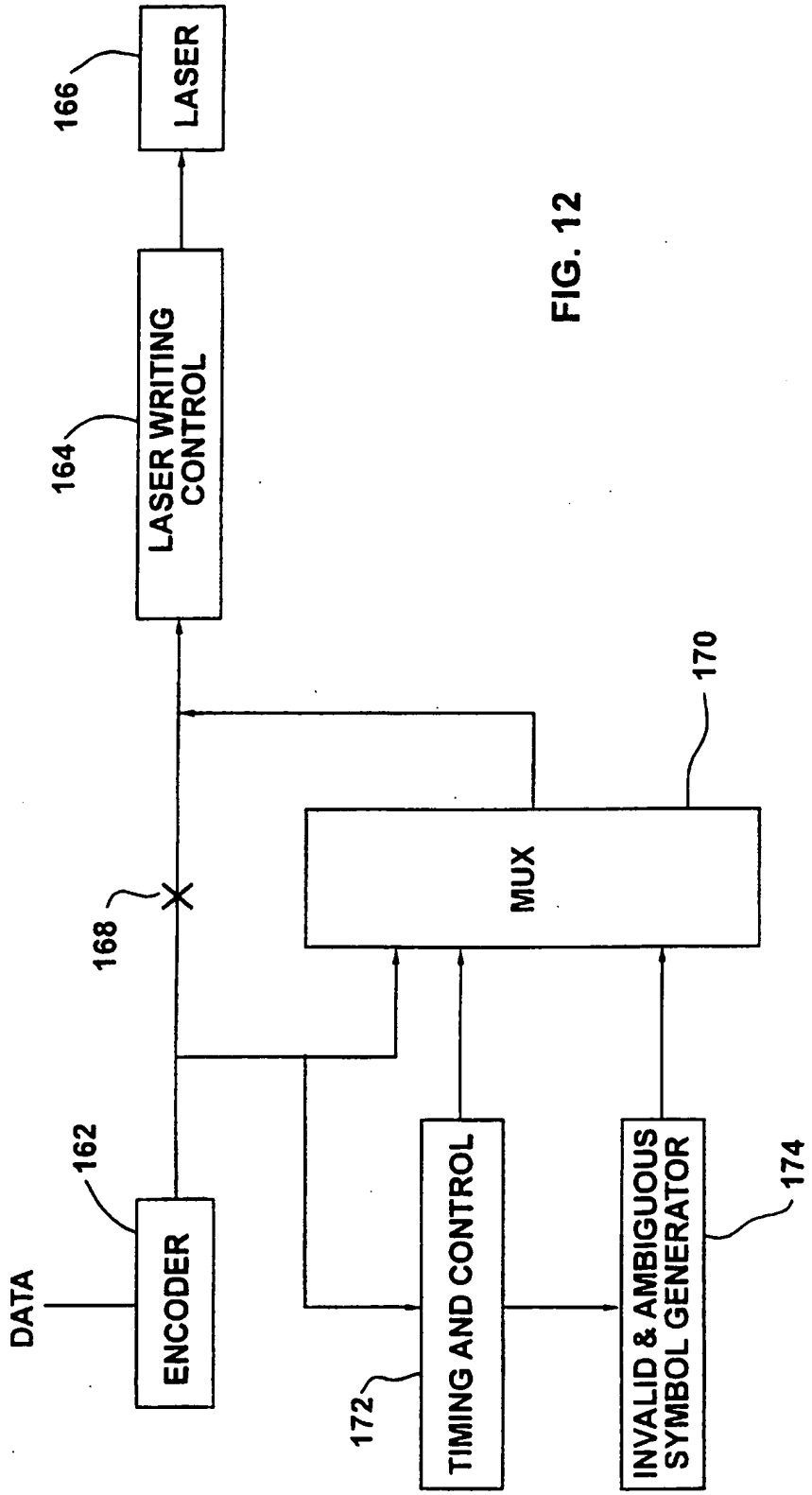

LASER
166
LASER WRITING CONTROL
164
MUX
170
168
ENCODER
162
160
DATA
TIMING AND CONTROL
172
INVALID & AMBIGUOUS SYMBOL GENERATOR
174

FIG. 12

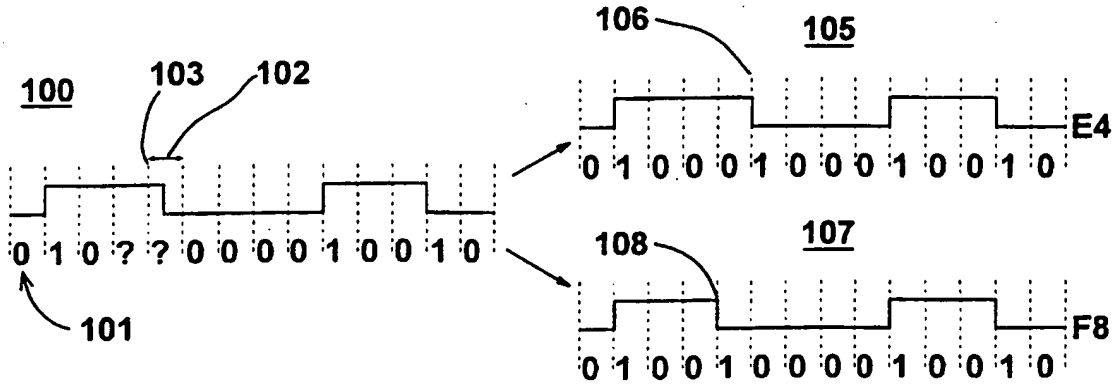

100
103
102
101
0 1 0 ? ? 0 0 0 0 1 0 0 1 0
106
105
E4
0 1 0 0 0 1 0 0 0 1 0 0 1 0
108
107
F8
0 1 0 0 1 0 0 0 0 1 0 0 1 0

FIG. 13

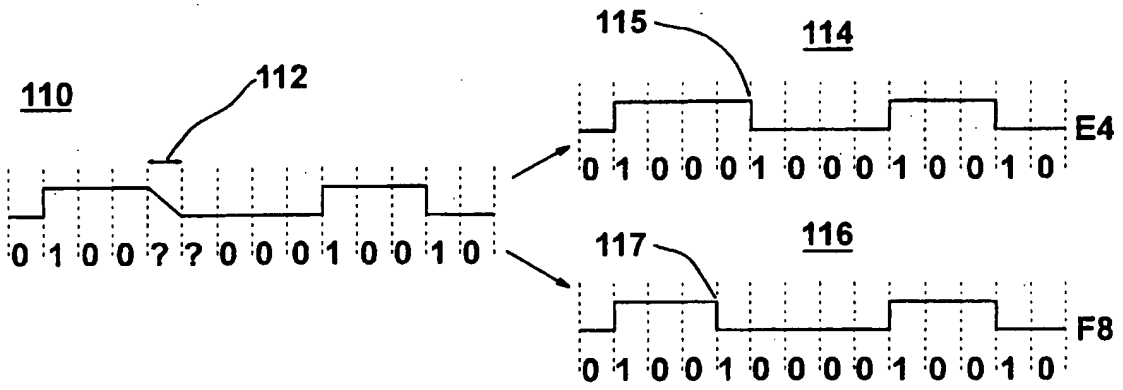

110
112
0 1 0 0 ? ? 0 0 0 1 0 0 1 0
115
114
E4
0 1 0 0 0 1 0 0 0 1 0 0 1 0
117
116
F8
0 1 0 0 1 0 0 0 0 1 0 0 1 0

FIG. 14

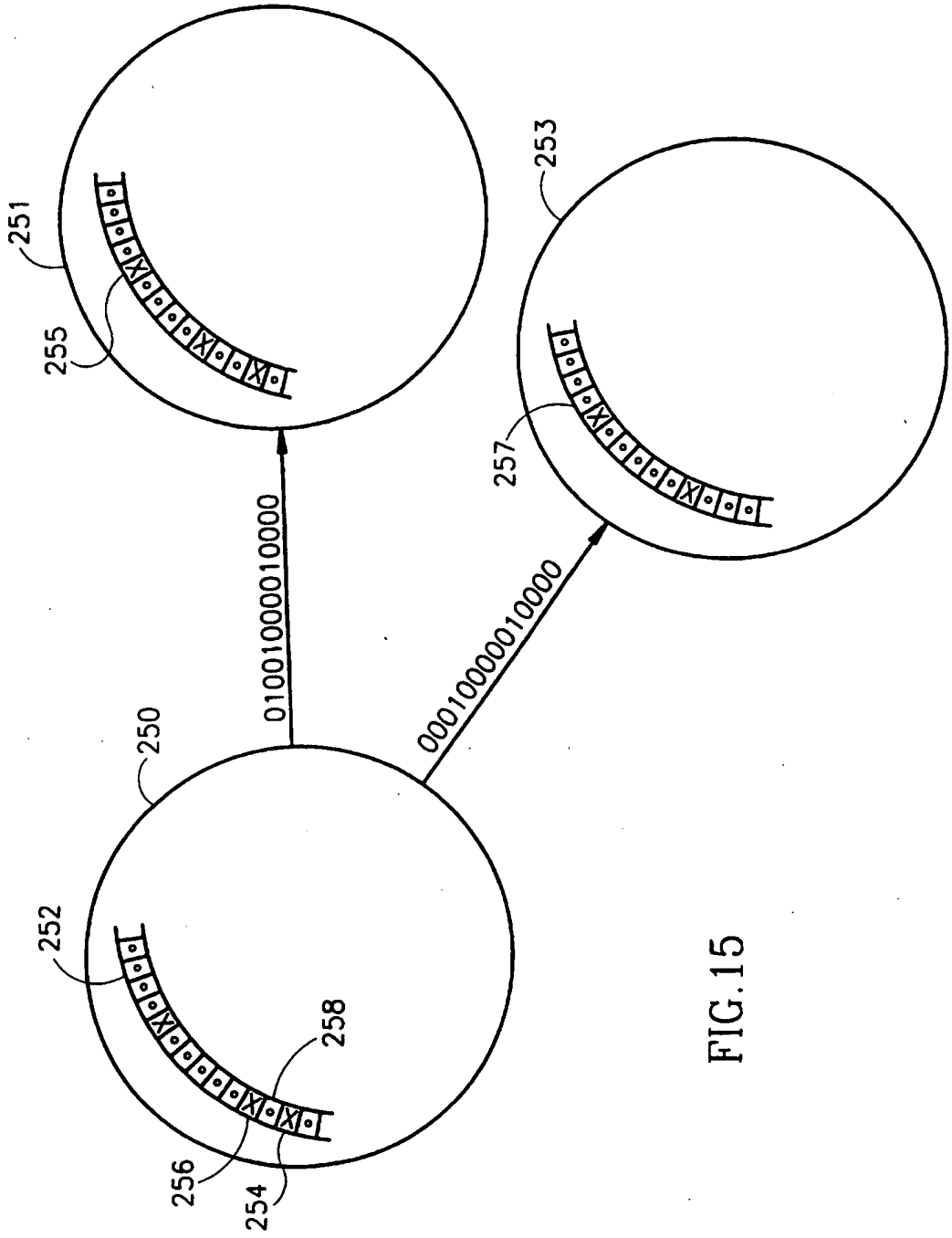

251
253
255
257
250
252
258
256
254
0100100001000
0001000001000

FIG. 15

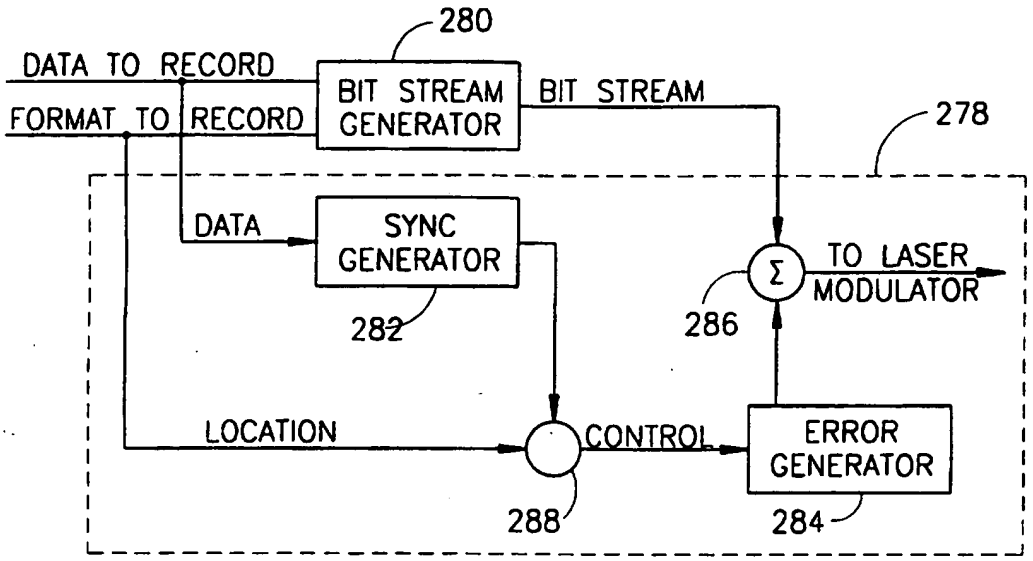

280
DATA TO RECORD
FORMAT TO RECORD
BIT STREAM GENERATOR
BIT STREAM
278
DATA
SYNC GENERATOR
282
Σ
TO LASER MODULATOR
286
LOCATION
CONTROL
ERROR GENERATOR
288
284

FIG.16

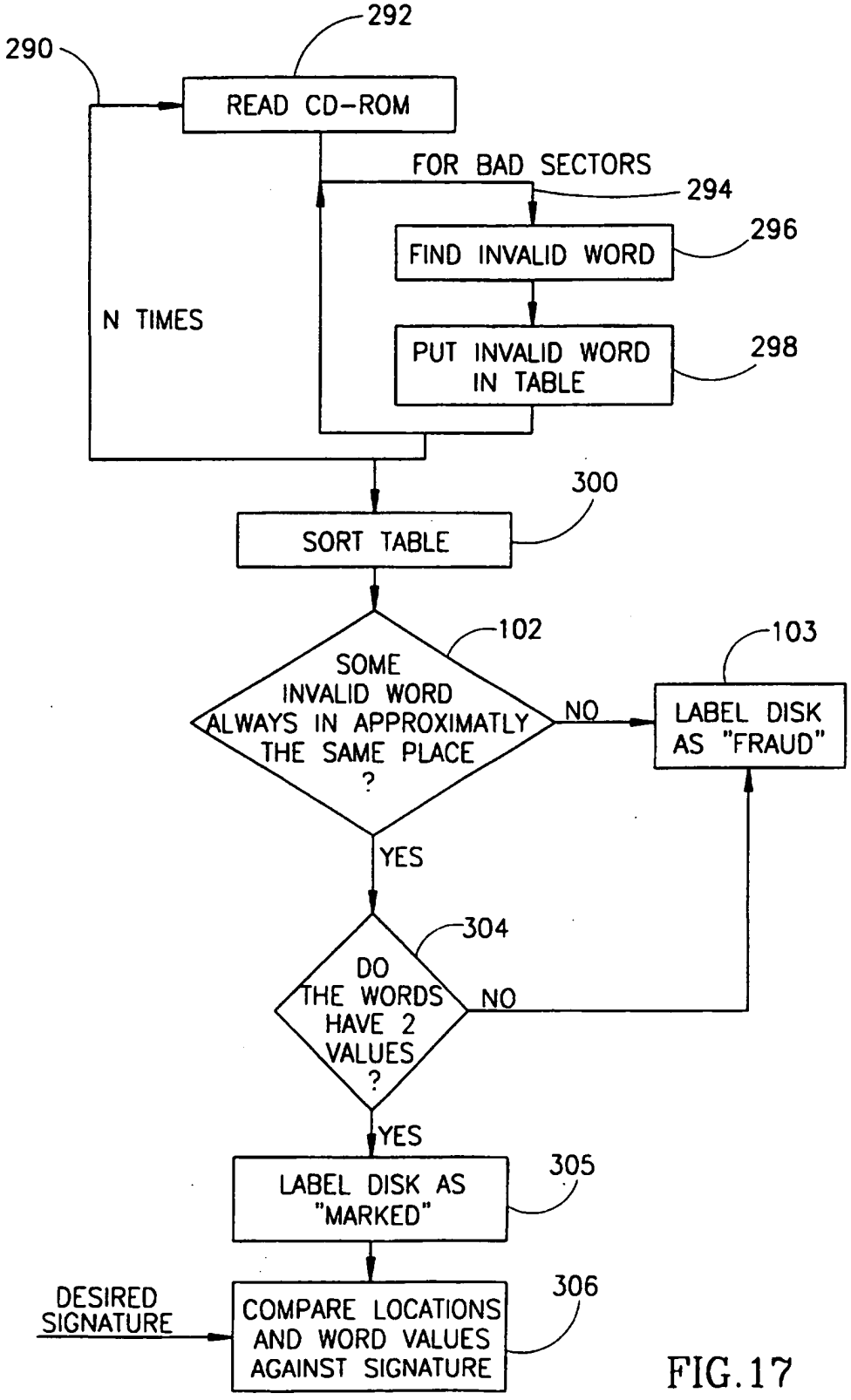

290
292
READ CD-ROM
FOR BAD SECTORS
294
FIND INVALID WORD
296
N TIMES
PUT INVALID WORD IN TABLE
298
300
SORT TABLE
102
SOME INVALID WORD ALWAYS IN APPROXIMATLY THE SAME PLACE ?
NO
103
LABEL DISK AS "FRAUD"
YES
304
DO THE WORDS HAVE 2 VALUES ?
NO
YES
LABEL DISK AS "MARKED"
305
DESIRED SIGNATURE
COMPARE LOCATIONS AND WORD VALUES AGAINST SIGNATURE
306

FIG.17

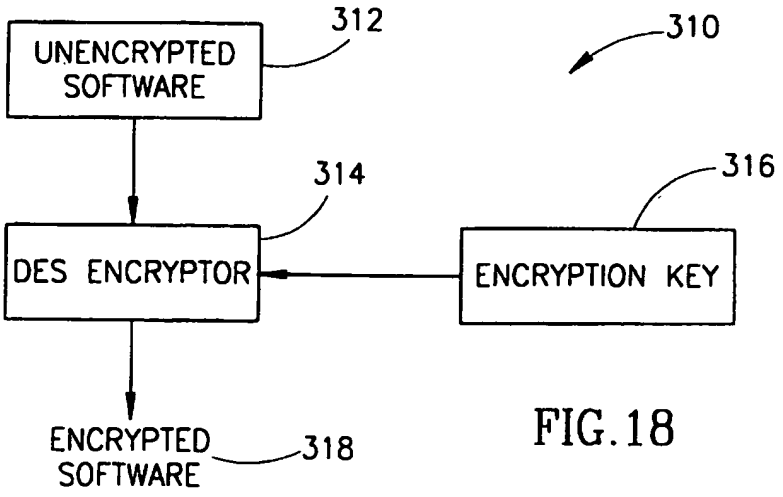

UNENCRYPTED SOFTWARE
312
310
314
316
DES ENCRYPTOR
ENCRYPTION KEY
ENCRYPTED SOFTWARE
318

FIG.18

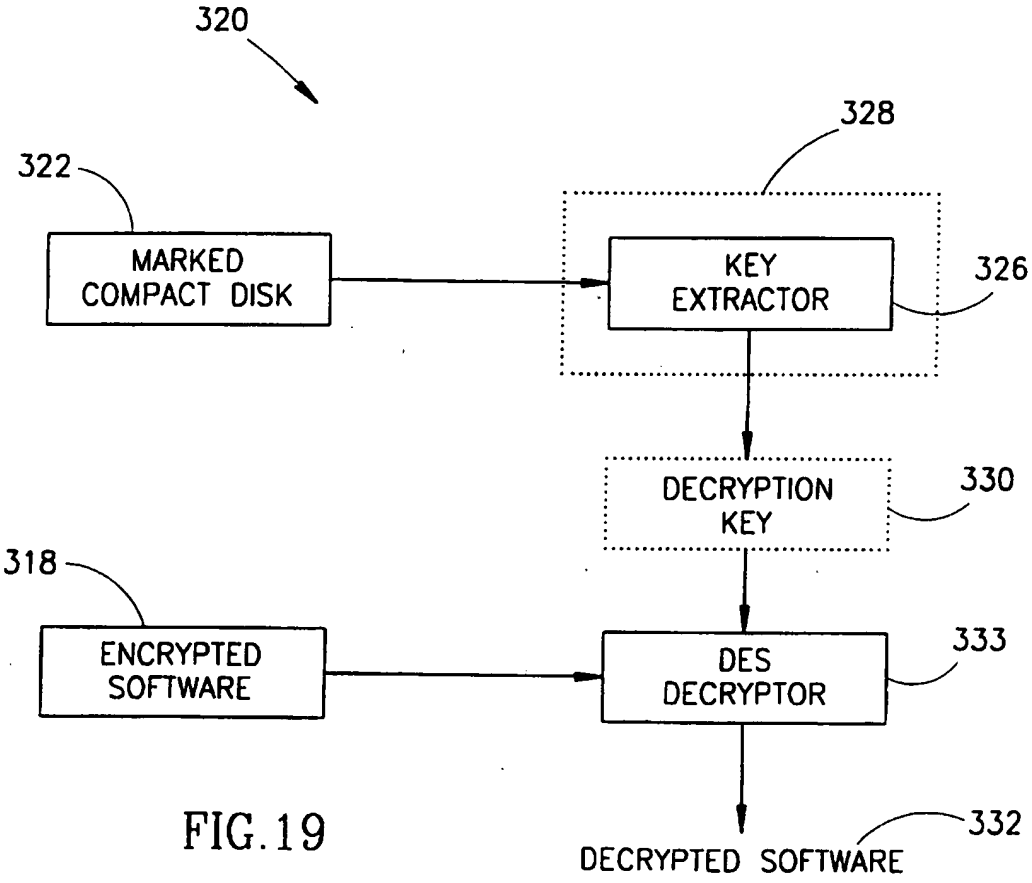

320
328
322
MARKED COMPACT DISK
KEY EXTRACTOR
326
DECRYPTION KEY
330
318
ENCRYPTED SOFTWARE
DES DECRYPTOR
333
DECRYPTED SOFTWARE
332

FIG.19